F. Switch less Communications Services The switchless network 2168 is a term used for the application of cell-switching or packet- switching techniques to both data and isochronous multimedia communications services. In the past, circuit switching was the only viable technology for transport of time-sensitive isochronous voice. Now, with the development of Asynchronous Transfer Mode cell switching networks which provide quality of service guarantees, a single network infrastructure which serves both isochronous and bursty data services is achievable.

The switchless network is expected to provide a lower cost model than circuit switched architectures due to: Flexibility to provide exactly the bandwidth required for each application, saving bandwidth when no data is being transferred. A minimum 56 Kbps circuit will not automatically be allocated for every call.

Adaptability to compression techniques, further reducing bandwidth requirements for each network session.

Lower costs for specialized resource equipment, due to the fact that analog ports do not have to be supplied for access to special DSP capabilities such as voice recognition or

conferencing. A single high-bandwidth network port can serve hundreds of"calls" simultaneously.

Applicability and ease of adaptation of the switches networks to advanced high-bandwidth services such as videoconferencing, training on demand, remote expert, integrated video/voice/fax/electronic mail, and information services. Figure 23 illustrates a sample switches network 2168 in accordance with a preferred embodiment.

G. Governing Principles 1. Architectural Principles This section contains a listing of architectural principles which provide the foundation of the architecture which follows.

Service Principles 1. The Service Model must support seamless integration of new and existing services.

2. Services are created from a common Service Creation Environment (SCE) which provides a seamless view of services

3. All services execute in common service logic execution environments (SLEEs), which do not require software changes when new services are introduced.

4. All services are created from one or more service features.

5. Data stored in a single customer profile in the ISP Data Servers may be used to drive multiple services.

6. The Service Model must support the specification and fulfillment of quality of service parameters for each service. These quality of service parameters, when taken together, constitute a service level agreement with each customer. Service deployment must take into account specified quality of service parameters.

2. Service Feature Principles 1. All service features are described by a combination of one or more capabilities.

2. All service features can be defined by a finite number of capabilities.

3. Individual service features must be defined using a standard methodology to allow service designers to have a common understanding of a capability. Each service

feature must document their inputs, outputs, error values, display behaviors, and potential service applications.

4. Interaction of physical entities in the network implementation shall not be visible to the user of the service feature through the service feature interfaces.

5. Each service feature should have a unified and stable external interface. The interface is described as a set of operations, and the data required and provided by each operation.

6. Service features are not deployed into the network by themselves. A service feature is only deployed as part of a service logic program which invokes the service feature (see Figure 21). Thus, service features linked into service logic programs statically, while capabilities are linked to service logic programs dynamically. This is where the loose coupling of resources to services is achieved.

3. Capability Principles 1. Capabilities are defined completely independent from consideration of any physical or logical implementation (network implementation independent).

2. Each capability should have a unified and stable interface. The interface is described as a set of operations, and the data required and provided by each operation.

3. Individual capabilities must be defined using a standard methodology to allow service designers to have a common understanding of a capability. Each capability must document their inputs, outputs, error values, display behaviors, and potential service applications.

4. Interaction of physical entities in the network implementation shall not be visible to the user of the capability through the capability interfaces.

5. Capabilities may be combined to form high-level capabilities.

6. An operation on a capability defines one complete activity. An operation on a capability has one logical starting point and one or more logical ending points.

7. Capabilities may be realized in one or more piece of physical hardware or software in the network implementation.

8. Data required by each capability operation is defined by the capability operation support data parameters and user instance data parameters.

9. Capabilities are deployed into the network independent of any service.

10. Capabilities are global in nature and their location need not be considered by the service designer, as the whole network is regarded as a single entity from the viewpoint of the service designer.

11. Capabilities are reusable. They are used without modification for other services.

4. Service Creation, Deployment, and Execution Principles 1. Each Service Engine 2134 supports a subset of the customer base. The list of customers supported by a service engine is driven by configuration data, stored on the ISP Data Server 2182.

2. Each Service Engine 2134 obtains its configuration data from the ISP data servers 2152 at activation time.

3. Service Engines 2134 use ISP database clients 2180 (see the data management section of this description) to cache the data necessary to support the customers configured for that service engine 2134, as needed. Caching can be controlled by the ISP database server 2182, or controlled by the database of the ISP database server 2182.

Data may be cached semi-permanently (on disk or in memory) at a service engine 2134 if it is deemed to be too much overhead to load data from the data server 2182 on a frequent basis.

4. Service Engines 2134 may be expected to execute all of a customer' s services, or only a subset of the customer's services. However, in the case of service interactions, one Service Engine 2134 must always be in control of the execution of a service at any given time. Service Engines may hand-off control to other service engines during the course of service execution.

5. Service Engines do not own any data, not even configuration data.

6. Service Engines 2134 are not targets for deployment of data. Data Servers 2182 are targets for deployment of data.

5. Resource Management Model 2150 Principles 1. Resources 2152 should be accessible from anywhere on the network.

2. Resources are not service-specific and can be shared across all services if desired.

3. Resources of the same type should be managed as a group.

4. The Resource Management Model 2150 should be flexible enough to accommodate various management policies, including: Least Cost, Round Robin, Least Recently Used, Most Available, First Encountered, Use Until Failure and Exclusive Use Until Failure.

5. The Resource Management Model 2150 should optimize the allocation of resources and, if possible, honoring a selected policy.

6. The RM 2150 must allow for a spectrum of resource allocation techniques ranging from static configuration to fully dynamic allocation of resources on a transaction by transaction basis.

7. The Resource Management Model 2150 must allow for the enforcement of resource utilization policies such as resource time out and preemptive reallocation by priority.

8. The Resource Management Model 2150 must be able to detect and access the status, utilization and health of resources in a resource pool.

9. All Resources 2152 must be treated as managed objects.

10. All resources must be able to register with the RM 2150 to enter a pool, and de- register to leave a pool.

11. The only way to request, acquire and release a resource 2152 is through the RM 2150.

12. The relationship between resources should not be fixed, rather individual instances of a given resource should be allocated from a registered pool in response to need or demand.

13. All specialized resources 2152 must be manageable from a consistent platform-wide viewpoint.

14. All specialized resources 2152 must offer SNMP or CMIP agent functionality either directly or through a proxy.

15. Every specialized resource 2152 shall be represented in a common management information base.

16. All specialized resources shall support a standard set of operations to inquire, probe, place in or out of service, and test the item.

17. All specialized resources shall provide a basic set of self-test capabilities which are controlled through the standard SNMP or CMIP management interfaces.

6. Data Management 2138 Principles 1. Multiple copies of any data item are allowed.

2. Multiple versions of the value of a data item are possible, but one view is considered the master.

3. Master versions of a given data item are under a single jurisdiction.

4. Multiple users are allowed to simultaneously access the same data.

5. Business rules must be applied uniformly across the ISP 2100 to ensure the validity of all data changes.

6. Users work on local copies of data; data access is location independent and transparent.

7. From the data management point of view, users are applications or other software components.

8. Data access should conform to a single set of access methods which is standardized across the ISP 2100.

9. Private data is allowed at a local database, but cannot be shared or distributed.

10. Only master data can be shared or distributed.

11. Private formats for a shared data item are allowed at the local database.

12. Transactional capabilities can be relaxed at end-user discretion if allowed within the business rules.

13. Rules-based logic and other meta-data controls provide a flexible means to apply policy.

14. Data Replication provides reliability through duplication of data sources.

15. Database Partitioning provides scalability by decreasing the size of any particular data store, and by decreasing the transaction rate against any particular data store.

16. Data Management 2138 must allow both static and dynamic configuration of data resources.

17. Common data models and common schemas should be employed.

18. Logical application views of data are insulated from physical data operations such as relocation of files, reloading of databases, or reformatting of data stores.

19. Audit trails, and event histories, are required for adequate problem resolutions.

20. On-line data audits and reconciliation are required to ensure data integrity.

21. Data recovery of failed databases is needed in real time.

22. Data metrics are needed for monitoring, trending, and control purposes.

23. 7 by 24 operation with 99.9999 availability is required.

24. Data Management 2138 mechanisms must scale for high levels of growth.

25. Data Management 2138 mechanisms must provide cost effective solutions for both large-scale and small-scale deployments.

26. Data Management mechanisms must handle overload conditions gracefully.

27. Data processing and data synchronization must occur in real-time to meet our business needs.

28. Trusted order entry and service creation should work directly on the ISP databases rather than through intermediary applications whenever possible.

29. All data must be protected; additionally customer data is private and must retain its confidentiality.

30. Configurations, operational settings, and run-time parameters are mastered in the ISP MIB (management information base).

31. Wherever possible, off the shelf data solutions should be used to meet Data Management needs.

The following principles are stated from an Object-oriented view: 32. Data items are the lowest set of persistent objects; these objects encapsulate a single data value.

33. Data items may have a user defined type.

34. Data items may be created and deleted.

35. Data items have only a single get and set method.

36. The internal value of a data item is constrained by range restrictions and rules

37. Data items in an invalid state should be inaccessible to users.

7. Operational Support Principles 1. Common View - All ISP 2100 Operational Support User Interfaces should have the same look & feel.

2. Functional Commonality - The management of an object is represented in the same manner throughout the ISP Operational support environment.

3. Single View - A distributed managed object has a single representation at the ISP Operational Support User Interfaces, and the distribution is automatically.

4. OS/DM Domain - Data within the Operational support domain should be managed with the ISP Data Management 2138 Mechanisms.

5. Global MIB - There is a logical Global MIB which represents resources in the entire ISP.

6. External MIBs - Embedded MIBs that are part of a managed component are outsider of Operational Support and Data Management. Such MIBs will be represented to the OS by a Mediation Device.

7. System Conformance - System conformance to the ISP OS standards will be gained through Mediation Layers.

8. Operational Functions - Operational personnel handle the Network Layer & Element Management for physical & logical resources.

9. Administration Functions - Administration personnel handle the Planning & Service Management.

10. Profile Domain - Service & customer profile data bases are managed by administration personnel under the domain of the Data Management system.

11. Telecommunication Management Network (TMN) compliance - TMN compliance will be achieved through a gateway to any TMN system.

12. Concurrent - Multiple Operators & Administrators must be able to simultaneously perform operations from the ISP OS Interfaces.

8. Physical Model Principles 1. Compatibility: The physical network model provides backward compatibility for existing telecommunications hardware and software.

2. Scaleable: The physical network model is scaleable to accommodate a wide range of customer populations and service requirements.

3. Redundant: The physical network model provides multiple paths of information flow across two network elements. Single points of failure are eliminated.

4. Transparent: Network elements is transparent to the underlying network redundancy.

In case of a failure, the switchover to redundant links is automatic.

5. Graceful Degradation: The physical network model is able to provide available services in a gradual reduction of capacity in the face of multiple network failures.

6. Interoperable: The physical network model allows networks with different characteristics to interoperate with different network elements.

7. Secure: The physical network model requires and provides secure transmission of information. It also has capabilities to ensure secure access to network elements.

8. Monitoring: The physical network model provides well-defined interfaces and access methods for monitoring the traffic on the network. Security (see above) is integrated to prevent unauthorized access to sensitive data.

9. Partitionable: The physical network model is (logically) partitionable to form separate administrative domains.

10. Quality of Service: The physical network model provides QOS provisions such as wide range of qualities, adequate QOS for legacy applications, congestion management and user-selectable QOS.

11. Universal Access: The physical network model does not prevent access to a network element due to its location in the network. A service is able to access any resource on the network.

12. Regulatory awareness: The physical network model is amenable at all levels to allow for sudden changes in the regulatory atmosphere.

13. Cost Effective: The physical network model allows for cost effective implementations by not being reliant on single vendor platforms or specific standards for function.

H. ISP Service Model This section describes the Service model of the Intelligent Services Platform Architecture Framework.

1. Purpose The ISP Service Model establishes a framework for service development which supports: rapid service creation and deployment; efficient service execution; complete customization control over services for customers; total service integration for a seamless service view for customers; improved reuse of ISP capabilities through loose coupling of those capabilities;

reduced cost of service implementation; and vendor-independence.

2. Scope of Effort The ISP Service Model supports all activities associated with Services, including the following aspects: provisioning; creation; deployment; ordering; updating; monitoring; execution; testing or simulation; customer support and troubleshooting; billing; trouble ticket handling; and operations support.

This model covers both marketable services and management services.

Marketable services are the services purchased by our customers Management services are part of the operation of the MCI network, and are not sold to customers.

The Service Model also defines interactions with other parts of the ISP Architecture, including Data Management, Resource Management, and Operational Support.

3. Service Model Overview Central to the Intelligent Services Platform is the delivery of Services 2200 (Figure 24).

Services are the most critical aspect in a telecommunication service provider's ability to make money. The following definition of services is used throughout this service model:

A service 2200 is a set of capabilities combined with well-defined logic structures and business processes which, when accessed through a published interface, results in a desired and expected outcome on behalf of the user.

One of the major differences between a Service 2200 and an Application 2176 or 2178 (Figure 22) is that a Service 2200 includes the business processes that support the sale.

operation, and maintenance of the Service. The critical task in developing a Service is defining what can be automated, and clearly delineating how humans interact with the Service.

4. Service Structure The vocabulary we will use for describing services includes the services themselves, service features, and capabilities. These are structured in a three-tier hierarchy as shown in Figure 24.

A service 2200 is an object in a sense of an object-oriented object as described earlier in the specification. An instance of a service 2200 contains other objects, called service features 2202. A service feature 2202 provides a well defined interface which abstracts the controlled interaction of one or more capabilities 2204 in the ISP Service Framework, on behalf of a service.

Service features 2202, in turn, use various capability 2204 objects. Capabilities 2204 are standard, reusable, network-wide building blocks used to create service features 2202. The key requirement in Service Creation is for the engineers who are producing basic capability objects to insure each can be reused in many different services as needed.

a) Services 2200 Services 2200 are described by "service logic," which is basically a program written in a very high-level programming language or described using a graphical user interface. These service logic programs identify: what service features 2202 are used; the order in which service features are invoked;

the source of input service data; the destination for output service data; error values and error handling; invocation of other services 2200; interaction with other services; and the interactions with other services; The service logic itself is generally not enough to execute a service 2200 in the network.

Usually, customer data is needed to define values for the points of flexibility defined in a service, or to customize the service for the customer's particular needs. Both Management and Marketable Services are part of the same service model. The similarities between of Management and Marketable Services allow capabilities to be shared. Also, Management and Marketable Services represent two viewpoints of the same network: Management Services represent and operational view of the network, and Marketable Services represent an external end-user or customer view of the network. Both kinds of services rely on network data which is held in common.

Every Marketable Service has a means for a customer to order the service, a billing mechanism, some operational support capabilities, and service monitoring capabilities. The Management Services provide processes and supporting capabilities for the maintenance of the platform.

b) Service Features 2202 Service features 2202 provide a well-defined interface of function calls. Service features can be reused in many different services 2200, just as capabilities 2204 are reused in many different service features 2202. Service features have specific data input requirements, which are derived from the data input requirements of the underlying capabilities. Data output behavior of a service feature is defined by the creator of the service feature, based upon the data available from the underlying capabilities. Service Features 2202 do not rely on the existence of any physical resource, rather, they call on capabilities 2204 for these functions, as shown in Figure 25.

Some examples of service features are: Time-based Routing - based on capabilities such as a calendar, date/time, and call objects, this feature allows routing to different locations based upon time.

Authentication - based upon capabilities such as comparison and database lookup, this function can be used to validate calling card use by prompting for a card number and/or an access number (pin number), or to validate access to a virtual private network.

Automated User Interaction - based upon capabilities such as voice objects (for recording and playback of voice), call objects (for transferring and bridging calls to specialized resources), DTMF objects (for collection or outpulsing of DTMF digits), vocabulary objects (for use with speech recognition), this feature allows automated interaction with the user of a service. This service feature object can be extended to include capabilities for video interaction with a user as well.

c) Capabilities 2204 A capability 2204 is an object, which means that a capability has internal, private state data, and a well-defined interface for creating, deleting, and using instances of the capability.

Invoking a capability 2204 is done by invoking one of its interface operations. Capabilities 2204 are built for reuse. As such, capabilities have clearly defined data requirements for input and output structures. Also, capabilities have clearly defined error handling routines.

Capabilities may be defined in object-oriented class hierarchies whereby a general capability may be inherited by several others.

Some examples of network-based capability objects are: voice (for recording or playback), call (for bridging, transferring, forwarding, dial-out, etc), DTMF (for collection or outpulsing), and Fax (for receive, send, or broadcast).

Some capabilities are not network-based, but are based purely on data that has been deployed into our platform. Some examples of these capabilities are: calendar (to determine what day of the week or month it is),

comparison (to compare strings of digits or characters) translation (to translate data types to alternate formats), and distribution (to choose a result based on a percentage distribution).

d) Service Data There are three sources for data while a service executes: Static Data defined in the service template, which include default values for a given service invocation.

Interactive Data obtained as the service executes, which may be explicit user inputs or derived from the underlying network connections.

Custom Data defined in User Profiles, which is defined by customers or their representatives when the service is requested (i.e. at creation time).

5. Service 2200 Execution Services 2200 execute in Service Logic Execution Environments (SLEEs). A SLEE is executable software which allows any of the services deployed into the ISP 2100 to be executed. In the ISP Architecture, Service Engines 2134 (Figure 21) provide these execution environments. Service Engines 2134 simply execute the services 2200 that are deployed to them.

Service templates and their supporting profiles are deployed onto database servers 2182 (Figure 22). When a SLEE is started on a Service Engine 2134, it retrieves its configuration from the database server 2182. The configuration instructs the SLEE to execute a list of services 2200. The software for these services is part of the service templates deployed on the database servers. If the software is not already on the Service Engine 2134, the software is retrieved from the database server 2182. The software is executed, and service 200 begins to run.

In most cases a service 2200 will first invoke a service feature 2202 (Figure 24) which allows the service to register itself with a resource manager 2188 or 2190. Once registered the service can begin accepting transactions. Next, a service 2200 will invoke a service feature

2202 which waits on an initiating action. This action can be anything from an internet logon, to an 800 call, to a point of sale card validation data transaction. Once the initiating action occurs in the network, the service select function 2148 (Figure 21) uses the Resource Manager 2150 function to find an instance of the executing service 2200 to invoke. The initiating action is delivered to the service 2200 instance, and the service logic (from the service template) determines subsequent actions by invoking additional service features 2202.

During service 2200 execution, profile data is used to determine the behavior of service features 2202. Depending on service performance requirements some or all of the profile data needed by a service may be cached on a service engine 2134 from the ISP 2100 database server 2182 to prevent expensive remote database lookups. As the service executes, information may generated by service features 2202 and deposited into the Context Database.

This information is uniquely identified by a network transaction identifier. In the case of a circuit-switched call, the already-defined Network Call Identifier will be used as the transaction identifier. Additional information may be generated by network equipment and deposited into the Context Database as well, also indexed by the same unique transaction identifier. The final network element involved with the transaction deposits some end-of- transaction information into the Context Database. A linked list strategy is used for determining when all information has been deposited into the Context Database for a particular transaction. Once all information has arrived, an event is generated to any service which has subscribed to this kind of event, and services may then operate on the data in the Context Database. Such operations may include extracting the data from the Context Database and delivering it to billing systems or fraud analysis systems.

6. Service Interactions In the course of a network transaction, more than one service can be invoked by

the network.

Sometimes, the instructions of one service may conflict with the instructions of another service. Here's an example of such a conflict: a VNET caller has a service which does not allow the caller to place international calls. The VNET caller dials the number of another VNET user who has a service which allows international dialing, and the called VNET user places an international call, then bridges the first caller with the international call. The original user was able to place an international call through a third party, in defiance of his

company's intention to prevent the user from dialing internationally. In such circumstances, it may be necessary to allow the two services to interact with each other to determine if operation of bridging an international call should be allowed.

The ISP service model must enable services 2200 to interact with other services. There are several ways in which a service 2200 must be able to interact with other services (see Figure 26): Transfer of Control 2210: where a service has completed its execution path and transfers control to another service; Synchronous Interaction 2212: where a service invokes another service and waits for a reply; Asynchronous Interaction 2214: where a service invokes another service, performs some other actions, then waits for the other service to complete and reply; or One Way Interaction 2216: where a service invokes another service but does not wait for a reply.

In the example of interacting VNET services above, the terminating VNET service could have queried the originating VNET service using the synchronous service interaction capability. The interesting twist to this idea is that service logic can be deployed onto both network-based platforms and onto customer premises equipment. This means that service interaction must take place between network-based services and customer-based services.

7. Service Monitoring Services 2200 must be monitored from both the customer's viewpoint and the network viewpoint. Monitoring follows one of two forms: The service 2200 can generate detailed event-by-event information for delivery to the transaction context database The service can generate statistical information for delivery periodically to a statistics database, or for retrieval on demand by a statistics database.

Analysis services can use the Statistics Database or the Context Database to perform real time or near real time data analysis services.

The Context Database collects all event information regarding a network transaction. This information will constitute all information necessary for network troubleshooting, billing, or network monitoring.

I. ISP Data Management Model This section describes the Data Management 2138 aspects of the Intelligent Services Platform (ISP) 2100 Target Architecture.

1. Scope The ISP Data Management 2138 Architecture is intended to establish a model which covers the creation, maintenance, and use of data in the production environment of the ISP 2100, including all transfers of information across the ISP boundaries.

The Data Management 2138 Architecture covers all persistent data, any copies or flows of such data within the ISP, and all flows of data across the ISP boundaries. This model defines the roles for data access, data partitioning, data security, data integrity, data manipulation, plus database administration. It also outlines management policies when appropriate.

2. Purpose The objectives of this architecture are to: Create a common ISP functional model for managing data; Separate data from applications; Establish patterns for the design of data systems; Provide rules for systems deployment; Guide future technology selections; and Reduce redundant developments and redundant data storage.

Additional goals of the target architecture are: Ensure data flexibility; Facilitate data sharing; Institute ISP-wide data control and integrity; Establish data security and protection;

Enable data access and use; Provide high data performance and reliability; Implement data partitioning; and Achieve operational simplicity.

3. Data management Overview In one embodiment, the Data Management Architecture is a framework describing the various system components, how the systems interact, and the expected behaviors of each component. In this embodiment data is stored at many locations simultaneously, but a particular piece of data and all of its replicated copies are viewed logically as a single item.

A key difference in this embodiment is that the user (or end-point) dictates what data is downloaded or stored locally.

a) Domains Data and data access are characterized by two domains 2220 and 2222, as shown in Figure 27. Each domain can have multiples copies of data within it. Together, the domains create a single logical global database which can span international boundaries. The key aspect to the domain definitions below is that all data access is the same. There is no difference in an Order Entry feed from a Call Processing lookup or Network side data update.

Central domain 2220 controls and protects the integrity of the system. This is only a logical portrayal, not a physical entity. Satellite domain 2222 provides user access and update capabilities. This is only a logical portrayal, not a physical entity.

b) Partitions In general, Data is stored at many locations simultaneously. A particular piece of data and all of its replicated copies are viewed logically as a single item. Any of these copies may be partitioned into physical subsets so that not all data items are necessarily at one site.

However partitioning preserves the logical view of only one, single database.

c) Architecture The architecture is that of distributed databases and distributed data access with the following functionality: Replication and Synchronization; Partitioning of Data Files; Concurrency Controls; Transactional Capability; and Shared common Schemas.

Figure 28 shows logical system components and high-level information flows. None of the components depicted is physical. Multiple instances of each occur in the architecture.

The elements in Figure 28 are: NETWK 2224 - external access to the ISP 2100 from the network side; SVC I/F 2226 -the network interface into ISP; SYSTMS 2228 - external application such as Order Entry; G/W 2230 - a gateway to the ISP 2100 for external applications; dbAppl 2232 - a role requiring data access or update capabilities; dbClient 2234- the primary role of the satellite domain; dbServer 2236- the primary role of the central domain; dbAdmin 2238- an administrative role for Data; dbMon 2240- a monitoring role; I/F Admin 2242 administrative role for interfaces; and Ops 2244- operations console.

d) Information Flow The flows depicted in Figure 28 are logical abstractions; they are intended to characterize the type of information passing between the logical components.

The flows shown above are: Rest - data requests to the ISP from external systems; Resp -responses from the ISP to external requests; Access - data retrieval by applications within the ISP; Updates -data updates from applications within ISP;

Evts, data related events sent to the monitor; Meas - data related metrics sent to the monitor; New Data - additions to ISP master data; Changed Data changes to ISP master data; Views - retrieving ISP master data; Subscriptions -asynchronous stream of ISP master data; Cache copies- a snapshot copy of ISP master data; Actions- any control activity; and Controls any control data.

e) Domain Associations In general the Satellite domains 2232 of Data Management 2138 encompass: ISP Applications; .External systems Network interfaces 2226 and system gateways 2230; and Database client (dbClient) 2234.

The Central domain for Data Management 2138 encompasses: Monitoring (dbMon) 2240; Administration (dbAdmin) 2238; and Database masters (dbServer) 2236 4. Logical Description The behavior of each Architecture component is described separately below: a) Data Applications (dbAppl) 2232 This includes any ISP applications which require database access. Examples are the ISN NIDS servers, and the DAP Transaction Servers, The applications obtain their required data from the dbClient 2234 by attaching to the desired databases, and providing any required policy instructions. These applications also provide the database access on behalf of the

external systems or network element such as Order Entry or Switch requested translations.

Data applications support the following functionality: Updates: allow an application to insert, update, or delete data in an ISP database.

Access requests allow an application to search for data, list multiple items select items from a list or set, or iterate through members of a set.

Events and Measurements are special forms of updates which are directed to the monitoring function (dbMon) 2240.

b) Data Management 2138 (1) Client Databases (dbClient) 2234 The dbClients represent satellite copies of data. This is the only way for an application to access ISP data. Satellite copies of data need not match the format of data as stored on the dbServer 2236.

The dbClients register with master databases (dbServer) 2236 for Subscriptions or Cache Copies of data. Subscriptions are automatically maintained by dbServer 2236, but Cache Copies must be refreshed when the version is out of date.

A critical aspect of dbClient 2234 is to ensure that data updates by applications are serialized and synchronized with the master copies held by dbServer 2236. However, it is just as reasonable for the dbClient to accept the update and only later synchronize the changes with the dbServer (at which time exception notifications could be conveyed back to the originating application). The choice to update in lock-step, or not, is a matter of application policy not Data Management 2138.

Only changes made to the dbServer master copies are forwarded to other dbClients.

If a dbClient 2234 becomes inactive or loses communications with the dbServer; it must resynchronize with the master. In severe cases, operator intervention may be required to reload an entire database or selected subsets.

The dbClient 2234 offers the following interface operations: Attach by an authorized application to a specified set of data; Policy preferences to be set by an authorized application; Select a specified view of the local copy of data; Insert, Update, or Delete of the local copy of data; Synchronize subscripted data with the dbServer; and Expiration notifications from dbServer for cached data.

Additionally, the dbClients submit Logs or Reports and signal problems to the monitor (dbMon) 2240.

(2) Data Masters (dbServer) 2236 The dbServers 2236 play a central role in the protection of data. This is where data is 'owned' and master copies maintained. At least two copies of master data are maintained for reliability. Additional master copies may be deployed to improve data performance.

These copies are synchronized in lock-step. That is each update is required to obtain a corresponding master-lock in order to prevent update conflicts. The strict implementation policies may vary, but in general, all master copies must preserve serial ordering of updates, and provide the same view of data and same integrity enforcement as any other master copy.

The internal copies of data are transparent to the dbClients 2234.

The dbServer 2236 includes the layers of business rules which describe or enforce the relationships between data items and which constrain particular data values or formats.

Every data update must pass these rules or is rejected. In this way dbServer ensures all data is managed as a single copy and all business rules are collected and applied uniformly.

The dbServer 2236 tracks when, and what kind of, data changes are made, and provides logs and summary statistics to the monitor (dbMon) 2240. Additionally these changes are forwarded to any active subscriptions and Cache-copies are marked out of date via expiration messages.

The dbServer also provides security checks and authorizations and ensures that selected items are encrypted before storage.

The dbServer supports the following interface operations: View selected data from dbServer; Subscribe to selected data from dbServer; Copy selected data into a cache-copy at a dbClient 2234; Refresh a dbClient cache with the current copy on demand; New data insertion across all dbServer copies of the master; Change data attributes across all dbServer copies; and Cancel previous subscriptions and drop cache-copies of data

(3) Data Administration (dbAdmin) 2238 Data Administration (dbAdmin) 2238 involves setting data policy, managing the logical and physical aspect of the databases, and securing and configuring the functional components of the Data Management 2138 domain. Data Management policies include security, distribution, integrity rules, performance requirements, and control of replications and partitions.

dbAdmin 2238 includes the physical control of data resources such as establishing data locations, allocating physical storage, allocating memory, loading data stores, optimizing access paths, and fixing database problems. dbAdmin 2238 also provides for logical control of data such as auditing, reconciling, migrating, cataloguing, and converting data.

The dbAdmin 2238 supports the following interface operations: Define the characteristics of a data type;

Create logical containers of given dimensions; .Relate two or more containers through an association; Constrain data values or relations through conditional triggers and actions; Place physical container for data in a given location; Move physical containers for data to new locations; Remove physical containers and their data; Load data from one container to another; Clear the data contents of a container; and Verify or reconcile the data contents of a container.

(4) Data Monitoring (dbMon) 2240 The dbMon 2240 represents a monitoring function which captures all data-related events and statistical measurements from the ISP boundary gateways, dbClients 2234 and dbServers 2236. The dbMon 2240 mechanisms are used to create audit trails and logs.

The dbMon typically presents a passive interface; data is fed to it. However monitoring is a hierarchical activity and further analysis and roll-up (compilation of data collected at intervals, such as every minute, into longer time segments, such as hours or days) occurs within dbMon. Additionally dbMon will send alerts when certain thresholds or conditions are met.

The rate and count of various metrics are used for evaluating quality of Service (QOS) , data performance, and other service level agreements. All exceptions and date errors are logged and flow to the dbMon for inspection, storage, and roll-up.

dbMon 2240 supports the following interface operations: Setting monitor controls, filters, and thresholds; Logging of data related activity; Reports of status, metrics, or audit results; and Signaling alarms, or alerts.

(5) Data Management operations (Ops) 2244 The Operations consoles (Ops) 2244 provide the workstation-interface for the personnel monitoring, administering, and otherwise managing the system. The Ops consoles provide access to the operations interfaces for dbMon 2240, dbAdmin 2238, and dbServer 2236 described above. The Ops consoles 2244 also support the display of dynamic status through icon based maps of the vanous systems, interfaces, and applications within the Data management domain 2138.

5. Physical Description This section describes the Data Management 2138 physical architecture. It describes how a set of components could be deployed. A generalized deployment view is shown in Figure 29.

In Figure 29: circles are used to represent physical sites, boxes or combined boxes are computer nodes, and functional roles are indicated by abbreviations.

The abbreviations used in Figure 29 are: OE - order entry systems 2250; GW - ISP gateway 2230; . APP - application (dbAppl) 2232; CL- a dbClient 2234, SVR- a dbServer 2236; ADM- a dbAdmin component 2238; MON- a dbMon component 2240; and Ops - operations console.

The functional roles of these elements were described above (see Logical Description of the Target Architecture) in connection with Figure 28.

Each of the sites shown in Figure 29 is typically linked with one or more of the other sites by wide area network (WAN) links. The exact network configuration and sizing is left to a detailed engineering design task. It is not common for a database copy to be distributed to the Order Entry (OE) sites 2251, however in this architecture, entry sites are considered equivalent to satellite sites and will contain the dbClient functionality.

On the network-side of the ISP 2100, Satellite sites 2252 each contain the dbClient 2234 too.

These sites typically operate local area networks (LANs). The dbClients act as local repositories for network or system applications such as the ISN operator consoles, ARUs, or NCS switch requested translations.

The Central sites 2254 provide redundant data storage and data access paths to the dbClients 2234. Central sites 2254 also provide roll-up monitoring (dbMon) functions although dbMon components 2240 could be deployed at satellite sites 2252 for increased performance.

The administrative functions are located at any desired operations or administration site 2254 but not necessarily in the same location as the dbMon. Administrative functions require the dbAdmin 2238, plus an operations console 2244 for command and control. Remote operations sites are able to access the dbAdmin nodes 2238 from wide-area or local-area connections. Each of the sites is backed-up by duplicate functional components at other sites and are connected by diverse, redundant links.

6. Technology Selection The following section describes the vanous technology options which should be

considered.

The Data Management 2138 architecture does not require any particular technology to operate; however different technology choices will impact the resulting performance of the system.

Figure 30 depicts a set of technologies which are able to provide a very-high performance environment. Specific application requirements will determine the minimum level of acceptable performance. Three general environments are shown.

In the upper part, a multi-protocol routed network 2260 connects external and remote elements with the central data sites. Administrative terminals, and smaller mid-range computers are shown, plus a high-availability application platform such as Order Entry.

In the center are large-scale high-performance machines 2262 with large data-storage devices; these would be typical of master databases and data processing, and data capture/tracking functions such as dbServer 2236 and dbMon 2240.

In the lower part of the diagram are local area processing and network interfaces 2264, such as the ISN operator centers or DAP sites.

7. Implementations While much is known of the current ISP data systems, additional detailed requirements are necessary before any final implementations are decided. These requirements must encompass existing ISN, NCS, EVS, NIA, and TMN system needs, plus all of the new products envisioned for Broadband, Internet, and Switchless applications.

8. Security ISP data is a protected corporate resource. Data access is restricted and authenticated. Data related activity is tracked and audited. Data encryption is required for all stored passwords, PINS (personal identification numbers), private personnel records, and selected financial, business, and customer information. Secured data must not be transmitted in clear-text forms.

9. Meta-Data Meta-data is a form of data which comprises the rules for data driven logic. Meta-data is used to describe and manage (i.e. manipulate) operational forms of data. Under this architecture, as much control as possible is intended to be driven by meta-data. Meta-data (or data-driven logic) generally provides the most flexible run-time options. Meta-data is typically under the control of the system administrators.

10. Standard Database Technologies Implementation of the proposed Data Management Architecture should take advantage of commercially available products whenever possible. Vendors offer database technology, replication services, Rules systems, Monitoring facilities, Console environments, and many other attractive offerings

J. ISP Resource Management Model This section describes the Resource Management 2150 Model as it relates to the ISP 2100 Architecture.

a) Scope The Resource Management Model covers the cycle of resource allocation and de-allocation in terms of the relationships between a process that needs a resource, and the resource itself.

This cycle starts with Resource Registration and De-registration and continues to Resource Requisition, Resource Acquisition, Resource Interaction and Resource Release.

b) Purpose The Resource Management 2150 Model is meant to define common architectural guidelines for the ISP development community in general, and for the ISP Architecture in particular.

c) Objectives In the existing traditional ISP architecture, services control and manage their own physical and logical resources. Migration to an architecture that abstracts resources from services requires defining a management functionality that governs the relationships and interactions between resources and services. This functionality is represented by the Resource Management 2150 Model.

The objectives of the Resource Management Model are designed to allow for network-wide resource management and to optimize resource utilization, to enable resource sharing across the network: Abstract resources from services; Provide real-time access to resource status; Simplify the process of adding and removing resources; Provide secure and simple resource access; and Provide fair resource acquisition, so that no one user of resources may monopolize the use of resources.

d) Background Concepts Generally, the Resource Management 2150 Model governs the relationships and interactions between the resources and the processes that utilize them. Before the model is presented, a solid understanding of the basic terminology and concepts used to explain the model should be established. The following list presents these terms and concepts:

(1) Definitions Resource: A basic unit of work that provides a specific and well-defined capability when invoked by an external process. Resources can be classified as logical, like a service engine and a speech recognition algorithm, or physical, like CPU, Memory and Switch ports. A resource may be Shared like an ATM link bandwidth or Disk space, or Dedicated like a VRU or a Switch port.

Resource Pool: A set of registered resource members that share common capabilities.

Service: A logical description of all activities and the interaction flow between the user of the network resources and the resources themselves.

Policy: A set of rules that governs the actions taken on resource allocation and de- allocation, resource pool size thresholds and resource utilization thresholds.

(2) Concepts The Resource Management Model is a mechanism which governs and allows a set of functions to request, acquire and release resources to/from a resource pool through well-defined procedures and policies. The resource allocation and de-allocation process involves three phases: Resource Requisition is the phase in which a process requests a resource from the Resource Manager 2150.

Resource Acquisition: If the requested resource is available and the requesting process has the privilege to request it, the Resource Manager 2150 will grant the resource and the process can utilize it. Otherwise, the process has the choice to either abandon the resource allocation process and may try again later, or it may request that the Resource Manager 2150 grant it the resource whenever it becomes available or within a specified period.

Resource Release: The allocated resource should be put back into the resource pool once the process no longer needs it. Based on the resource type, the process either releases the resource and the resource informs the Resource Manager of its new

status, or the process itself informs the Resource Manager that the resource is available. In either case, the Resource Manager will restore the resource to the resource pool.

The Resource Management Model allows for the creation of resource pools and the specification of the policies governing them. The Resource Management Model allows resources to register and de-register as legitimate members of resource pools.

Resource Management Model policies enforce load balancing, failover and least cost algorithms and prevent services from monopolizing resources. The Resource Management Model tracks resource utilization and automatically takes corrective action when resource pools are not sufficient to meet demand. Any service should be able to access and utilize any available resource across the network as long as it has the privilege to do so.

The Resource Management Model adopted the OSI Object Oriented approach for modeling resources. Under this model, each resource is represented by a Managed Object (MO). Each MO is defined in terms of the following aspects: Attributes: The attributes of a MO represent its properties and are used to describe its characteristics and current states. Each attribute is a associated with a value, for example the value CURRENT~STATE attribute of a MO could be IDLE.

Operations: Each MO has a set of operations that are allowed to be performed on it.

These operations are: Create: to create a new MO Delete: to delete an existing MO Action: to perform a specific operation such as SHUTDOWN.

Get Value: to obtain a specific MO attribute value Add Value: to add specific MO attribute value Remove Value: to delete a specific MO attribute value from a set of values.

Replace Value: to replace an existing MO attribute value(s) with a new one.

Set Value: to set a specific MO attribute to its default value.

Notification: Each MO can report or notify its status to the management entity. This could be viewed as triggers or traps.

Behavior: The behavior of an MO is represented by how it reacts to a specific operation and the constraints imposed on this reaction. The MO may react to either external stimuli or internal stimuli. An external stimuli is represented by a message that carries an operation. The internal stimuli, however, is an internal event that occurred to the MO like the expiration of a timer. A constraint on how the MO should react to the expired timer may be imposed by specifying how many times the timers has to expire before the MO can report it.

All elements that need to utilize, manipulate or monitor a resource need to treat it as a MO and need to access it through the operations defined above. Concerned elements that need to know the status of a resource need to know how to receive and react to events generated by that resource.

Global and Local Resource Management: The Resource Management Model is hierarchical with at least two levels of management: Local Resource Manager (LRM) 2190 and Global Resource Manager (GRM) 2188. Each RM, Local and Global, has its own domain and functionality.

2. The Local Resource Manager (LRM): Domain: The domain of the LRM is restricted to a specific resource pool (RP) that belongs to a specific locale of the network. Multiple LRMs could exist in a single locale, each LRM may be responsible for managing a specific resource pool.

Function: The main functionality of the LRM is to facilitate the resource allocation and de-allocation process between a process and a resource according the Resource Management Model guidelines.

3. The Global Resource Manager (GRM) 2188: Domain: The domain of the GRM 2188 covers all registered resources in all resource pools across the network.

Function: The main function of the GRM is to help the LRM 2190 locate a resource that is not available in the LRM domain.

Figure 31 illustrates the domains of the GRM 2188 and LRM 2190 within network 2270.

4. The Resource Management Model (RMM) The Resource Management Model is based on the concept of Dynamic Resource Allocation as opposed to Static Configuration. The Dynamic Resource Allocation concept implies that there is no pre-defined static relationship between resources and the processes utilizing them.

The allocation and de-allocation process is based on supply and demand. The Resource Managers 2150 will be aware of the existence of the resources and the processes needing resources can acquire them through the Resource Managers 2150. On the other hand, Static Configuration implies a pre-defined relationship between each resource and the process that needs it. In such a case, there is no need for a management entity to manage these resources.

The process dealing with the resources can achieve that directly. Dynamic Resource Allocation and Static Configuration represent the two extremes of the resource management paradigms. Paradigms that fall between these extremes may exist.

The Resource Management Model describes the behavior of the LRM 2190 and GRM 2188 and the logical relationships and interactions between them. It also describes the rules and policies that govern the resource allocation and de-allocation process between the LRM/GRM and the processes needing the resources.

a) Simple Resource Management Model Realizing that resource allocation and de-allocation could involve a complex process, a simple form of this process is presented here as an introduction to the actual model. Simple resource allocation and de-allocation is achieved through six steps. Figure 32 depicts these steps.

1. A process 2271 requests the resource 2173 from the resource manager 2150.

2. The resource manager 2150 allocates the resource 2173.

3. The resource manager 2150 grants the allocated resource 2173 to the requesting process 2271.

4. The process 2271 interacts with the resource 2273.

5. When the process 2271 is finished with the resource 2273 it informs the resource.

6. The resource 2273 releases itself back to the resource manager 2150.

b) The Resource Management Model Logical Elements: The Resource Management Model is represented by a set of logical elements that interact and co-operate with each other in order to achieve the objectives mentioned earlier. These elements are shown in Figure 33 and include: Resource Pool (RP) 2272, LRM 2190, GRM 2188 and Resource Management Information Base (RMIB) 2274.

(1) Resource Pool (RP) 2272 All resources that are of the same type, share common attributes or provide

the same capabilities, and are located in the same network locale may be logically grouped together to form a Resource Pool (RP) 2272. Each RP will have its own LRM 2190.

(2) The Local Resource Manager (LRM) 2190 The LRM 2190 is the element that is responsible for the management of a specific RP 2272.

All processes that need to utilize a resource from a RP that is managed by a LRM should gain access to the resource through that LRM and by using the simple Resource Management Model described above.

(3) The Global Resource Manager (GRM) 2188 The GRM 2188 is the entity that has a global view of the resource pools across the network.

The GRM gains this global view through the LRMs 2190. All LRMs update the GRM with RP 2272 status and statistics. There are cases where a certain LRM can not allocate a resource because all local resources are busy or because the requested resource belongs to another locale. In such cases, the LRM can consult with the GRM to locate the requested resource across the network.

(4) The Resource Management Information Base (RMIB) 2274 As mentioned above, all resources will be treated as managed objects (MO). The RMIB 2274 is the database that contains all the information about all MOs across the network. MO information includes object definition, status, operation, etc. The RMIB is part of the ISP Data Management Model. All LRMs and the GRM can access the RMIB and can have their own view and access privileges of the MO's information through the ISP Data Management Model.

5. Component Interactions To perform their tasks, the Resource Management Model elements must interact and co- operate within the rules, policies and guidelines of the Resource Management Model. The following sections explain how these entities interact with each other.

a) Entity Relationship (ER) Diagram (Figure 33): In Figure 33, each rectangle represents one entity, the verb between the "o" implies the relationship between two entites and the square brackets "[i" imply that the direction of the relationship goes from the bracketed number to the non bracketed one. The numbers imply is the relationship is l-to-l, l-to-many or many-to-many.

Figure 33 can be read as follows: 1. One LRM 2190 manages one RP 2272.

2. Many LRMs 2190 access the RMIB 2274.

3. Many LRMs 2190 access the GRMs 2188.

4. Many GRMs 2188 access the RMIB 2274.

b) Registration and De-registration Resource registration and de-registration applies only on the set of resources that have to be dynamically managed. There are some cases where resources are statically assigned.

LRMs 2190 operate on resource pools 2272 where each resource pool contains a set of resource members. In order for the LRM to manage a certain resource, the resource has to inform the LRM of its existence and status; Also, the GRM 2188 needs to be aware of the availability of the resources across the network in order to be able to locate a certain resource.

The following registration and de-registration guidelines should be applied on all resources that are to be dynamically managed: All resources must register to their LRM 2190 as members of a specific resource pool 2272.

All resources must de-register from their LRM 2190 if, for any reason, they need to shutdown or be taken out of service.

All resources must report their availability status to their LRM 2190.

All LRMs must update the GRM 2188 with the latest resource availability based on the registered and de-registered resources.

c) GRM, LRM and RP Interactions Every RP 2272 will be managed by an LRM 2190. Each process that needs a specific resource type will be assigned an LRM that will facilitate the resource access. When the process needs a resource it must request it through its assigned LRM. When the LRM receives a request for a resource, two cases may occur: 1. Resource is available: In this case, the LRM allocates a resource member of the pool and passes a resource handle to the process. The process interacts with the resource until it is done with it. Based on the resource type, once the process is done with the resource, it either informs the resource that it is done with it, and the resource itself informs its LRM that it is available, or it releases the resource and informs the LRM that it is no longer using the resource.

2. Resource is not available: In this case, the LRM 2190 consults with the GRM 2188 for an external resource pool that contains the requested resource. If no external resource is available, the LRM informs the requesting process that no resources are available. In this case, the requesting process may: give up and try again, request that the LRM allocate the resource whenever it becomes available, or

request that the LRM allocates the resource if it becomes available within a specified period of time.

If an external resource is available, the GRM 2188 passes location and access information to the LRM 2190. Then the LRM either: allocates the resource on the behalf of the requesting process and passes a resource handle to it (In this case the resource allocation through the GRM is transparent to the process), or advises the requesting process to contact the LRM that manages the located resource.

d) GRM, LRM and RMIB Interactions The RMIB 2274 contains all information and status of all managed resources across the network. Each LRM 2190 will have a view of the RMIB 274 that maps to the RP 2272 it manages. The GRM 2188, on the other hand, has a total view of all resources across the network. This view consists of all LRMs views. The GRM's total view enables it to locate resources across the network.

In order for the RMIB 2274 to keep accurate resource information, each LRM 2190 must update the RMIB with the latest resource status. This includes adding resources, removing resources and updating resource states.

Both the LRM 2190 and GRM 2188 can gain their access and view of the RMIB 2274 through the ISP Data Management entity. The actual management of the RMIB data belongs to the ISP Data Management entity. The LRM and GRM are only responsible for updating the RMIB.

K. Operational Support Model 1. Introduction Most of the existing ISP service platforms were developed independently, each with it's own set of Operational Support features. The amount of time required to learn how to operate a given set of platforms increases with the number of platforms. The ISP service platforms need to migrate to an architecture with a common model for all of its Operational Support

features across all of its products. This requires defining a model that will support current needs and will withstand or bend to the changes that will occur in the future. The Operational Support Model (OSM) defines a framework for implementation of management support for the ISP 2100.

a) Purpose The purpose of the Operational Support Model is to: achieve operational simplicity by integrating the management platform for ISP resources; reduce the learning curve for operational personnel by providing a common management infrastructure; reduce the cost of management systems by reducing overlapping management system development; improve time to market for ISP services by providing a common management infrastructure for all of the ISP services and network elements; and provide a framework for managing ISP physical resources (hardware) and logical resources (software).

b) Scope The OSM described here provides for the distributed management of ISP physical network elements and the services that run on them. The management framework described herein could also be extended to the management of logical (software) resources. However, the architecture presented here will help map utilization and faults on physical resources to their resulting impact on services.

The management services occur within four layers Planning, Service Management, Network Layers, and Network Elements.

Information within the layers falls into four functional areas: Configuration Management, Fault Management, Resource Measurement, and

Accounting.

The use of a common Operational Support Model for all of the ISP will enhance the operation of the ISP, and simplify the designs of future products and services within the ISP.

This operational support architecture is consistent with the ITU Telecommunications Management Network (TMN) standards.

c) Definitions Managed Object: A resource that is monitored, and controlled by one or more management systems Managed objects are located within managed systems and may be embedded in other managed objects. A managed object may be a logical or physical resource, and a resource may be represented by more than one managed object (more than one view of the object).

Managed System: One or more managed objects.

Management Sub-Domain: A Management domain that is wholly located within a parent management domain.

Management System: An application process within a managed domain which effects monitoring and control functions on managed objects and/or management sub-domains.

Management Information Base : A MIB contains information about managed objects.

Management Domain: A collection of one or more management systems, and zero or more managed systems and management sub-domains.

Network Element: The Telecommunications network consist of many types of analog and digital telecommunications equipment and associated support equipment, such as transmission systems, switching systems, multiplexes, signaling terminals, front-end processors, mainframes, cluster controllers, file servers, LANs, WANs, Routers, Bridges, Gateways, Ethernet Switches, Hubs, X.25 links, SS7 links, etc. When managed, such equipment is generally referred to as a network element (NE).

Domain: The management environment may be partition in a number a ways such as functionally (fault, service...), geographical, organizational structure, etc.

Operations Systems: The management functions are resident in the Operations System.

2. The Operational Support Model Figure 34 shows the four management layers 2300, 2302, 2304 and 2306 of the Operational Support Model 2308 over the network elements 2310. The Operational Support Model 2308 supports the day to day management of the ISP 2100. The model is organized along three dimensions. Those dimensions are the layers 2300-2306, the functional area within those layers, and the activities that provide the management services. Managed objects (a resource) are monitored, controlled, and altered by the management system.

a) The Functional Model The following sections describe the functional areas as they occur within the management layers 2300-2306.

(1) Planning The ISP Planning Layer 2300 is the repository for data collected about the ISP 2100, and the place where that data is to provide additional value.

Configuration Management 2312: Setting of policy, and goals.

Fault Management 2314: Predicting of mean time to failure.

Resource Measurement 2316: Predicting future resource needs (trending, capacity service agreement compliance, maintenance agreement, work force).

Accounting: Determine cost of providing services in order to support service pricing decisions.

(2) Service Management The Service Ordering, Deployment, Provisioning, Quality of Service agreements, and Quality of service monitoring are in the ISP Service Management layer 2302. Customers will have a restricted view of the SM layer 2302 to monitor and control their services. The SM layer provides a manager(s) that interacts with the agents in the NLMs. The SM layer also provides an agent(s) that interacts with the manager(s) in the Planning layer 2300. Managers within the SM layer may also interact with other managers in the SM layer. In that case there are manager-agent relationships at the peer level.

Configuration Management 2320: Service Definition, Service Activation, Customer Definition, Customer Activation, Service Characteristics, Customer Characteristics, hardware provisioning, software provisioning, provisioning of other data or other resources.

Fault Management 2322: Monitor and report violations of service agreement. Testing.

Resource Measurement 2324: Predict the violation of a service agreement and flag potential resource shortages. Predict the needs of current and future (trending) services.

Accounting 2326: Process and forward Accounting information.

Network Layer Management: The ISP Network Layer Management (NLM) Layer 2304 has the responsibility for the management of all the network elements, as presented by the Element Management, both individually and as a set. It is not concerned with how a particular element provides services internally. The NLM layer 2304 provides a manager(s) that interacts with the agents in the EMs

2306. The NLM layer also provides an agent(s) that interacts with the manager(s) in the SM layer 2302. Managers within the NLM layer 2304 may also interact other managers in the NLM layer. In that case there are manager agent relationships at the peer level.

Configuration Management 2328 provides functions to define the characteristics of the local and remote resources and services from a network wide perspective.

Fault Management 2330 provides functions to detect, report, isolate, and correct faults that occur across multiple NEs.

Resource Measurement 2332 provides for the network wide measurement, analysis, and reporting of resource utilization from a capacity perspective.

Accounting 2334 consolidates Accounting information from multiple sources.

(3) Element Management The Element Management Layer 2306 is responsible for the NEs 2310 on an individual basis and supports an abstraction of the functions provided by the NEs The EM layer 2306 provides a manager(s) that interact with the agents in the NEs. The EM layer also provides an agent(s) that interact with the manager(s) in the NLM layer 2304. Managers within the EM layer 2306 may also interact other managers in the EM layer. In that case there are manager agent relationships at the peer level.

Configuration Management 2336 provides functions to define the characteristics of the local and remote resources and services.

Fault Management 2338 provides functions to detect, report, isolate, and correct faults.

Resource Measurement 2340 provides for the measurement, analysis, and reporting of resource utilization from a capacity perspective.

Accounting 2342 provides for the measurement and reporting of resource utilization from an accounting perspective.

b) Network Element The computers, processes, switches, VRUs, internet gateways, and other equipment that provide the network capabilities are Network Elements 2310. NEs provide agents to perform operations on the behalf of the Element Management Layer 2306.

c) Information Model Figure 35 shows manager agent interaction. Telecommunications network management is a distributed information application process. It involves the interchange of management information between a distributed set of management application processes for the purpose of monitoring and controlling the network resources (NE) 2310. For the purpose of this exchange of information the management processes take on the role of either manager 2350 or agent 2352. The manager 2350 role is to direct management operation requests to the agent 2352, receive the results of an operation, receive event notification, and process the received information. The role of the agent 2352 is to respond to the manager's request by performing the appropriate operation on the managed objects 2354, and directing any responses or notifications to the manager. One manager 2350 may interact with many agents 2352, and the agent may interact with more than one manager. Managers may be cascaded in that a higher level manager acts on managed objects through a lower level manager. In that case the lower level manager acts in both manager and agent roles.

3. The Protocol Model a) Protocols The exchange of information between manager and agent relies on a set of communications protocols. TMN, which offers a good model, uses the Common Management Information Services (CMIS) and Common Management Information Protocol (CMIP) as defined in Recommendations X.710, and X.711. This provides a peer-to-peer communications protocol based on ITU's Application Common Service Element (X.217 service description & X.227 protocol description) and Remote Operation Service Element (X.219 service description & X.229 protocol description). FTAM is also supported as an upper layer protocol for file transfers. The use of these upper layer protocols is described in Recommendation X.812.

The transport protocols are described in Recommendation X.811. Recommendation X.811

also describes the interworking between different lower layer protocols. This set of protocols is referred to as Q3.

b) Common context In order to share information between processes there needs to be a common understanding of the interpretation of the information exchanged. ASN.l (X.209) with BER could be used to develop this common understanding for all PDU exchanged between the management processes (manager/agent).

c) Services of the upper layer The following identifies the minimum services required of the service layer and is modeled after the TMN CMIS services.

SET: To add, remove, or replace the value of an attribute.

GET: To read the value of an attribute.

CANCEL-GET:To cancel a previously issued GET.

ACTION: To request an object to perform a certain action.

CREATE: To create an object.

DELETE: To remove an object.

EVENT-REPORT: Allows the network resource to announce an event.

4. The Physical Model Figure 36 shows the ISP 2100 physical model.

5. Interface Points Mediation Device 2380 provides conversion from one information model to the ISP information model. Gateways 2382 are used to connect to management systems outside of the ISP. These gateways will provide the necessary functions for operation with both ISP compliant systems, and non-compliant systems. The gateways may contain mediation devices 2380. Figure 36 identifies nine interface points. The protocols associated with those interface points are:

1. There are two upper layer protocols. The protocol for communications with the workstation and the ISP upper layer for all other operational support communications. The lower layer is TCP/IP over Ethernet.

2. The upper layer is the protocol for communications with workstation 2384, and the lower layer is TCP/IP over Ethernet.

3,4. The upper layer is the ISP upper layer, and the lower layer is TCP/IP over Ethernet.

5. The proprietary protocols are the of legacy systems that are not compatible with the supported interfaces. Equipment that provides a Simple Network Management Protocol (SNMP) interface will be supported with Mediation Devices.

6,7,8,9. Gateways by their nature will support ISP compliant and non-compliant interfaces.

Gateways to enterprise internal systems could include such as the Order Entry system, or an enterprise wide TMN system.

The ISP Realization of the Operational Support Model Figure 37 shows operational support realization

6. General The Operational Support Model provides a conceptual framework for building the Operational Support System. Figure 37 represents an ISP realization of this conceptual model. In this implementation of that model all the ISP Network Elements would be represented to the Operational Support System by a Management Information Base (MIB) 2370 and the agent process that acts upon the objects in the MIB.

Field support personnel have two levels from which the ISP 2100 will be managed.

1. For trouble-shooting, the Network Layers Manager 2372 gives field support a picture of the ISP as a whole. The process of detecting, isolating, and correcting problems begins from

there. From that layer, problems could be isolated to a single Network Element. Individual Network Elements are accessible from the Network Element Managers 2374 and would allow a more detailed level of monitoring, control, configuration, and testing. The centralized view of the ISP is missing from today's ISP, but many recognize its importance.

For configuration the Network Layers Manager 2370 provides an ISP-wide view, and interacts with the Network Element Managers 2374 to configure Network Elements in a consistent manner. This will help insure that the ISP configuration is consistent across all platforms. The ability to change a piece of information in one place and have it automatically distributed ISP- wide is a powerful tool that has not been possible with the current ISP management framework.

Once a service definition has been created from the Service Creation Environment 2376, the Service Manager 2378 is used to place it in the ISP network, and provision the network for the new service. Customers for a service are provisioned through the Service Manager 2378.

As a part of provisioning customers the Service Manager predicts resource utilization, and determines if new resources need to be added to handle the customer's use of a service. It uses the current utilization statistics as a basis for that determination. Once a customer is activated, the Service Manager monitors the customer S usage of the service to determine if the quality of service agreement is being met. As customer utilization of the services increases the Service Manager 2378 predicts the need to add resources to the ISP network.

This Service Management, with appropriate restrictions, can be extended to customers as another service. While Service Creation is the talk of the IN world, it needs a Service Manager that is integrated with the rest of the system, and that is one of the purposes of this model.

Finally, for planning personnel (non-field support), the Planning Manager 2380 analyzes the ISP-wide resource utilization to determine future needs, and to allocate cost to different services to determine the cost of a service as the basis for future service pricing.

L. PhysicalNehwork Model 1. Introduction This section describes the Physical Network aspects of the Intelligent Services Platform (ISP) 2100 Architecture.

a) Purpose The Physical Network Model covers the: Logical Architecture Mapping; Information Flows; and Platform Deployment in the production environment of the architecture.

b) Scope This model defines the terminology associated with the physical network, describes the interactions between various domains and provides examples of realizations of the architecture.

c) Objectives The objectives of this model are to: Create a model for identifying various network platforms; Classify Information Flow; Provide standard nomenclature; Provide rules for systems deployment; and Guide future technology selections.

2. Information Flow One of the key aspects of the intelligent network (IN) is the Information Flow across various platfonns installed in the network. By identifying types of information and classifying them, the network serves the needs of IN.

Customers interact with IN in a series of call flows. Calls may be audio-centric (as in the conventional ISP products), multimedia-based (as in internetMCI user using the web browser), video-based (as in video-on-demand) or a combination of contents.

Information can be classified as follows: .Content; Signaling; or Data.

Normally, a customer interacting with the intelligent network will require all three types of information flows.

a) Content Content flows contain the primary information being transported. Examples of this are analog voice, packet switched data, streamed video and leased line traffic. This is customer's property that IN must deliver with minimum loss, minimum latency and optimal cost. The IN elements are standardized such that the transport fabric supports more connectivity suites, in order to allow content to flow in the same channels with flow of other information.

b) Signaling Signaling flows contain control information used by network elements. ISUP RLT/IMT, TCP/IP domain name lookups and ISDN Q.931 are all instances of this. The IN requires, uses and generates this information. Signaling information coordinates the various network platforms and allows intelligent call flow across the network. In fact, in a SCE-based IN, service deployment will also require signaling information flowing across the fabric.

c) Data Data flows contain information produced by a call flow, including crucial billing data records often produced by the fabric and certain network platforms.

3. Terminology Network: A set of interconnected network elements capable of transporting content, signaling and/or data. MCI's IXC switch fabric, the ISP extended WAN, and the Internet backbone are classic examples of networks. Current installations tend to carry different contents on different networks, each of which is specialized for specific content transmission.

Both technology and customer requirements (for on-demand high bandwidth) will require carriers to use more unified networks for the majority of the traffic. This will require the fabric to allow for different content characteristics and protocols along the same channels.

Another aspect of this will be more uniform content-independent signaling.

Site: A set of physical entities collocated in a geographically local area. In the current ISP architecture, instances of sites are Operator Center, ISNAP Site (which also has ARU's) and an EVS site. By the very definition, the NT and DSC switches are NOT part of the site. They are instead part of the Transport Network (see below). In the architecture, a group of (geographically collocated) Service Engines (SE), Special Resources (SR), Data Servers (DS) along with Network Interfaces and Links form a site.

Network Element: A physical entity connecting to the Transport Networks through Network Interfaces. Examples of this are ACP, EVS SIP, MTOC, Videoconference Reservation Server, DAP Transaction Server, and NAS. In the next few years, elements such as web servers, voice authentication servers, video streamers and network call record stores will join the present family of network elements.

Network Interface: Equipment enabling connectivity of Network Elements to the Transport Networks. DS 1 CSU/DSU, 1 Base Ethernet interface card and ACD ports are network interfaces. With the architecture of the preferred embodiment, network interfaces will provide a well-understood uniform set of API's for communication.

Link: Connection between 2 or more Network Interfaces which are at different sites. A link may be a segment of OC-12 SONET Fiber or 100mbps dual ring FDDI section. In the coming years, IN must handle network links such as ISO Ethernet WAN hub links and gigabit rate OC-48's.

Connection: an attachment of two or more Network Interfaces which are at the same site.

Figure 38 shows a representation of a physical network 2400 schematic. Networks 2401 contain network elements 2402 at sites 2404 are interconnected through network interfaces 2406 and one or more gateways 2408.

4. Entity Relationships Entity relationships as shown in Figure 39 have been arrived at as part of the physical network modeling rules. Some of these rules allow for generalities that future demands and some will constrain definitions to avoid conflicts.

1. A Network 2401 spans one or more sites 2404, and contains one or more network elements 2402.

2. A Site 2404 contains one or more network elements 2402.

3. A Network Element 2402 is located in only one Site 2404.

4. A Link 2420 connects two or more Sites 2404.

5. A Connection 2422 connects two or more Network Elements.

6. A Network Element 2402 contains one or more Network Interfaces 2406.

The preferred embodiment integrates product and service offerings for MCI's business customers. The initial embodiment focuses on a limited product set. Requirements for an interface have been identified to capitalize on the integration of these services. The interface provides user-manageability of features, distribution list capabilities, and a centralized message database.

VIII. INTELLIGENT NETWORK All of the platform's support services have been consolidated onto one platform. The consolidation of platforms enables shared feature/functionality of services to create a common look and feel of features.

A. Network Management The architecture is designed such that it can be remotely monitored by an MCI operations support group. This remote monitoring capability provides MCI the ability to: Identify degraded or broken connectivity between: -platforms, servers or nodes that must pass information (i.e., objects) to the "universal inbox", -platforms, servers or nodes responsible for retrieving messages and delivering messages,

-the "universal inbox and the PC Client messaging interface, -the "universal inbox " and the Message Center interface, -platforms, servers or nodes that must pass profile information to Profile, and -platforms, servers or nodes that must pass profile information to the ARU; Identify degraded application processes and isolate the process that is degraded; Identify hardware failure; and Generate alarms that can be detected and received by an internal MCI monitoring group for all application process, hardware or interface failures.

In addition, remote access to system architecture components is provided to the remote monitoring and support group such that they can perform remote diagnostics to isolate the cause of the problem.

B. Customer Service Customer Service teams support all services. Customer support is provided to customers in a seamless manner and encompasses the complete product life cycle including: Alpha tests;

Beta tests; Commercial release; and Identification of enhancements to address customer feedback or additional customer support requirements Comprehensive and coordinated support procedures ensure complete customer support from inception to termination. Customer service is provided from the time the Account Team submits the order until the customer cancels the account. Comprehensive and coordinated customer support entails the following: A one-stop, direct access, customer service group to support ARU or VRU problems, WWW Browser problems or PC Client problems.

A staff that is well trained on diagnosing problems associated with access (ARU, WWW Browser or PC Client), the user interface (ARU, WWW Browser or PC Client), the application (Message Center or Profile Management) or the back-end system interfaces (universal inbox, directlineMCI voicemail/faxmail platform, Fax Broadcast System, SkyTel Paging server, order entry systems, billing systems, etc.)

A staff that has on-line access to databases with information about ARU or VRU capabilities, WWW Browser capabilities, identified hardware issues and identified application issues 7 x 24 customer support a single toll free number (800 or 888) with direct access to the customer service group seamless first, second and third level support for most troubles where: - Level 1 support is the first support representative answering the telephone. They are expected to be able to resolve the most commonly asked questions or problems reported by customers. These questions or problems typically deal with access type (ARU, WWW Browser, PC Client), dial-up communication for the WWW Browser or PC Client, installation or basic computer (PC, workstation, terminal) hardware questions. Additionally they are able to open and update trouble tickets, and reactivate customers' passwords.

- Level 2 support is provided within the customer support group when referrals to more experienced technical experts is necessary.

- Level 3 support may involve an outside vendor for on-site hardware support for the customer or an internal MCI engineering or support group depending on the nature of the problem. The customer support group will be able to track the status of the customer visit and add the identified problem to both the customer support databases.

- Level 4 support will continue to be provided by the Systems Engineering programmers.

Staffing levels to provide acceptable customer hold times and abandon rates.

A staff that has on-line access to the order entry and billing systems.

Automatically generate weekly reports that detail volume of calls made, received, average hold-time of calls and number of trouble tickets opened/closed/escalated.

C. Accounting Accounting is supported according to current MCI procedures.

D. Commissions Commissions are supported according to current MCI procedures.

E. Reporting Reporting is required for revenue tracking, internal and external customer installation/sales, usage and product/service performance. Weekly and monthly fulfillment reports are required from the fulfillment house(s). These fulfillment reports correlate the number of orders received and number of orders delivered. In addition, reporting identifies the number of different subscribers accessing Profile Management or the Message Center through the WWW Site.

F. Security Security is enforced in accordance with MCI's published policies and procedures for Internet security. In addition, security is designed into the WWW Browser and ARU interface options to verify and validate user access to directlineMCI profiles, Message Center, Personal Home Page calendars and Personal Home Page configurations.

G. Trouble Handling Trouble reporting of problems is documented and tracked in a single database. All troubles are supported according to the Network Services Trouble Handling System (NSTHS) guidelines. Any Service Level Agreements (SLAs) defined between MCI organizations are structured to support NSTHS

Any troubles that require a software fix are closed in the trouble reporting database and opened as a Problem Report (PR) in the Problem Tracking System. This Problem Tracking System is used during all test phases of and is accessible by all engineering and support organizations.

IX. ENHANCED PERSONAL SERVICES Throughout this description, the following terms will be used: Term Represents Server Both the hardware platform and a TCP service

Web Server AIX 4.2 system running Netscape Commerce Server HTTP Daemon Welcome Server Application Server The Web Servers running as Welcome Servers will be running the Netscape

Commerce Server HTTP Daemon in secure as well as normal mode. The Web Servers operating as various application servers will run this daemon in secure mode only. The Secure Mode uses SSLv2.

A. Web Server Architecture The Web Servers are located in a DMZ. The DMZ houses the Web Servers and associated Database Clients as required. The database clients do not hold any data, but provide an interface to the data repositories behind the corporate firewall.

The Web space uses Round-Robin addressing for name resolution. The Domain name is registered with the administrators of mci.com domain, with a sub-netted (internally autonomous) address space allocated for galileo.mci.com domain.

Figure 40 shows the sequence of events leading to a successful login.

1. Welcome Server 450 This Web Server runs both the secure and normal HTTP daemons. The primary function of this server is to authenticate user 452 at login time. The authentication requires the use of Java and a switch from normal to secure mode operation. There are one or more Welcome servers 450 in the DMZ. The information provided by the Welcome server 450 is stateless.

The statelessness means that there is no need to synchronize multiple Welcome Servers 450.

The Welcome servers first task is to authenticate the user. This requires the use of single use TOKENS, Passcode authentication and Hostile IP filtering. The first is done using a Token Server 454, while the other two will be done using direct database 456 access.

In case of failed authentication the user 452 is shown a screen that mentions all the reasons (except Hostile-IP) why the attempt may have failed. This screen automatically leads the users back to the initial login screen.

Welcome server 450's last task, after a successful authentication, is to send a service selection screen to the user 452. The Service Selection screen directs the user to an appropriate Application Server. The user selects the Application, but an HTML file in the Server Section page determines the Application Server. This allows the Welcome Servers 450 to do rudimentary load balancing.

All the Welcome Servers 450 in the DMZ are mapped to www.galileo.mci.com. The implementation of DNS also allows galileo.mci.com to map to www.galileo.mci.com.

2. Token Server 454 This is a database client and not a Web Server. The Token servers 454 are used by Welcome Servers 450 to issue a TOKEN to login attempts. The issued TOKEN, once validated, is used to track the state information for a connection by the Application Servers. The TOKEN information is be maintained in a database on a database server 456 (repository) behind the corporate firewall.

The Token Servers 454 do the following tasks: 1. Issue single use TOKEN during authentication phase.

2. Validate single use TOKEN (mark it for multi use).

3. Validate multi-use TOKEN.

4. Re-validate multi-use TOKEN.

The Token Servers 454 are required to issue a unique TOKEN on every new request. This mandates a communication link between multiple Token Servers in order to avoid conflict of TOKEN values issued. This conflict is eliminated by assigning ranges to each Token Server 454.

The TOKEN is a sixteen character quantity made up of 62 possible character values in the set [0-9A-Za-z]. The characters in positions 0,1 and 2 for each TOKEN issued by the Token

Server are fixed. These character values are assigned to each Token Server at configuration time. The character at position 0 is used as physical location identifier. The character at position 1 identifies the server at the location while the character at position 2 remains fixed at '0'. This character could be used to identify the version number for the Token Server.

The remaining 13 characters of the TOKEN are generated sequentially using the same 62 character set described above. At startup the TOKEN servers assign the current system time to the character positions 15-10, and set positions 9-3 to '0'. The TOKEN values are then incremented sequentially on positions 15-3 with position 3 being least significant. The character encoding assumes the following order for high to low digit values : 'z'-'a', 'Z'-'A', '9'- '0'.

The above scheme generates unique tokens if the system time is computed in 4 byte values, which compute to 6 base-62 characters in positions 15-10. The other assumption is that the scheme does not generate more than 62^7 (35*10A12) TOKENS in one second on any given Token Server in any embodiment.

The use of TOKEN ranges allows the use of multiple Token Servers in the Domain without any need for explicit synchronization. The method accommodates a maximum 62 sites. each having no more than 62 Token Servers. An alternate embodiment would accommodate more sites.

All of the Token Servers in the DMZ are mapped to token.galileo.mci.com. The initial embodiment contains two Token Servers 454. These Token Servers 454 are physically identical to the Welcome Servers 450, i.e., the Token Service daemon will run on the same machine that also runs the HTTP daemon for the Welcome service. In another embodiment, the two run on different systems.

The Welcome Server(s) 450 use the Token Server(s) 454 to get a single use TOKEN during the authentication phase of the connection. Once authenticated, the Welcome Server 450 marks the TOKEN valid and marks it for multiple use. This multi-use TOKEN accompanies the service selection screen sent to the user by the Welcome Server.

The design of TOKEN database records is discussed in detail below.

3. Application Servers The Application servers are Web servers that do the business end of the user transaction. The Welcome Server's last task, after a successful authentication, is to send a service selection screen to the user. The service selection screen contains the new multi-use TOKEN.

When the user selects a service, the selection request, with its embedded TOKEN, is sent to the appropriate Application Server. The Application Server validates the TOKEN using the Token Server 454 and, if valid, serves the request. A Token Server can authenticate a TOKEN issued by any one of the Token Servers on the same physical site. This is possible because the Token Servers 454 are database clients for the data maintained on a single database repository behind the corporate firewall.

An invalid TOKEN (or a missing TOKEN) always leads to the "Access Denied" page. This page is served by the Welcome Server(s) 450. All denial of access attempts are logged.

The actual operation of the Application Server depends on the Application itself. The Application Servers in the DMZ are mapped to <appName><num>.galileo.mci.com. Thus, in an embodiment with multiple applications (e.g., Profile Management, Message Center, Start Card Profile, Personal Web Space etc.), the same Welcome and Token servers 450 and 454 are used and more Applications servers are added as necessary.

Another embodiment adds more servers for the same application. If the work load on an application server increases beyond its capacity, another Application Server is added without any changes to existing systems. The SERVERS and TOKEN~HOSTS databases (described below) are updated to add the record for the new server. The <num> part of the host name is used to distinguish the Application Servers.

There is no need to use DNS Round-robin on these names. The Welcome server 450 uses a configuration table (The SERVERS database loaded at startup) to determine the Application Server name prior to sending the service selection screen.

B. Web Server System Environment All the Web servers run the Netscape Commerce Server HTTP daemon. The Welcome Servers 450 run the daemon in normal as well as secure mode, while the Application Servers only run the secure mode daemon.

The Token Server(s) run a TCP service that runs on a well known port for ease of connection from within the DMZ. The Token Service daemon uses tcp~wrapper to deny access to all systems other than Welcome and Application server(s). In order to speed this authentication process, the list of addresses is loaded by these servers at configuration time, instead of using reverse name mapping at every request. The use of tcp~wrapper also provides the additional tools for logging Token Service activity.

The Application servers mostly work as front-ends for database services behind the firewall.

Their main task is to validate the access by means of the TOKEN, and then validate the database request. The database requests are to Create, Read, Update or Delete exiting records or data fields on behalf of the user. The Application Servers do the necessary validation and authority checks before serving the request.

1. Welcome Servers The Welcome Servers serve the HTML pages described below to the user at

appropriate times. The pages are generated using Perl-based Common Gateway Interface (CGI) scripts.

The Scripts reside in a directory which is NOT in the normal document-root directory of the HTTP daemon. The normal precautions regarding disabling directory listing and removing all backup files etc. are taken to ensure that CGI scripts are not readable to the user. Figure 41 shows the directory structure 455 on the Welcome Server 450.

Figure 41 shows that the <document~root> 456 is separated from the <server~root> 458. It also shows that the <document~root> directory holds only the welcome and access failure HTML pages.

The HTTP Server maps all requests to the "cgi" directory 460 based on the URL requested.

The CGI scripts use the HTML templates from the "template" directory 462 to create and send the HTML output to the users on fly.

The use of the URL to map to a CGI script out of the <document~root> 456 blocks access to the <document~root> directory 456 by a malicious user. Since every access to the Welcome Server 450 maps to a CGI script in the cgi directory 460 of the Welcome Server 450 security is ensured by calling the authentication function at start of every script.

The user Authentication libraries are developed in Perl to authenticate the user identity.

NSAPI's authentication phase routines also add features for TOKEN verification and access mode detection in the servers themselves

The Welcome Servers 450 read their operating parameters into their environment from the database 456 at startup. It is necessary to keep this information in the common database in order to maintain the same environment on multiple Welcome Servers 450.

a) Welcome Page The welcome page is sent as the default page when the Welcome Server 450 is first accessed.

This is the only page that is not generated using a cgi script, and it is maintained in the <document~root> directory 456. This page does the following: Confirms that the browser can display Frames. If the browser fails to display Frames correctly, this page will display an appropriate error message and direct the user to down load Microsoft Internet Explorer V3.0 or later.

Confirms that the browser can run Java. A failure will result in the user being directed to Microsoft Internet Explorer V3.0 or later.

If the browser successfully displays Frames and runs Java, then this page will automatically request the Welcome Server 450 to send a login page.

The last action by the Welcome page is done using the Java applet embedded in page. This also switches the user's browser from normal to secure mode.

b) Login Page The Login Page is a cgi-generated page that contains an embedded single use TOKEN, a Java applet, and form fields for the user to enter a User Id and Passcode. The page may display a graphic to emphasize service.

The processing of this page is padded to introduce an artificial delay. In the initial embodiment, this padding is set to zero.

The response from this page contains the TOKEN, a scrambled TOKEN value generated by the applet, User Id and Passcode. This information is sent to the Welcome server using a POST HTTP request by the Java applet. The POST request also contains the Applet signature.

If the login process is successful the response to this request is the Server Selection page. A failure at this stage results in an Access Failed page.

c) Server Selection Page The Server Selection Page is a cgi-generated page which contains an embedded multi-use TOKEN. This page also shows one or more graphics to indicate the types of services available to the user. Some services are not accessible by our users. In other embodiments, when more than one service exists, a User Services Database keyed on the User Id is used to generate this page.

The Welcome server uses its configuration information to embed the names of appropriate Application Servers with the view to sharing the load among all available Application Servers. This load sharing is done by using the configuration data read by the Welcome Server(s) during startup.

The Welcome Server selects an Application Server based upon entries in its configuration file for each of the services. These entries list the names of Application Server(s) for each application along with their probability of selection. This configuration table is loaded by the Welcome Servers at startup.

d) Access Failed Page The Access Failed Page is a static page. That displays a message indicating that the login failed because of an error in User Id, Passcode or both. This page automatically loads the Login Page after a delay of 15 seconds.

e) Access Denied Page The Access Denied Page is a static page that displays a message indicating that an access failed due to authentication error. This page automatically loads the Login Page after a delay of 15 seconds. The Access Denied page is called by the Application Servers when their authentication service fails to recognize a TOKEN. All loads of this page will be logged and monitored.

2. Token Servers 454 The TOKEN service on the Web site is the only source of TOKEN generation and authentication. The Tokens themselves are stored in a shared Database 456. This database can be shared among all Token servers. The Token Database is behind the firewall out of the DMZ.

The Token service provides the services over a well-known (>1024) TCP port. These services are provided only to a trusted host. The list of trusted hosts is maintained in a configuration database. This database is also maintained behind the firewall outside of the DMZ. The Token servers read their configuration database only on startup or when they receive a signal to refresh. The Token services are: Grant a single use TOKEN for login attempt.

Validate a single use TOKEN.

Validate a TOKEN.

Re-Validate a TOKEN.

TOKEN aging is implemented by a separate service to reduce the work load on the Token servers.

All access to the Token Server(s) is logged and monitored. The Token Service itself is written using the tcp~wrapper code available from MCI's internal security groups.

3. Profile Management Application Servers The profile management application server(s) are the only type of Application servers implemented in the first embodiment. These servers have the same directory layout as the Welcome Servers. This allows the same system system to be used for both services if necessary.

C. Security The data trusted by subscribers to the Web server is sensitive to them. They would like to protect it as much as possible. The subscribers have access to this sensitive information via the Web server(s). This information may physically reside on one or more database servers, but as far as the subscribers are concerned it is on Server(s) and it should be protected.

Presently only the following information needs to be protected in an embodiment: In other embodiments, profile information for directline account additional information is protected, including Email, Voice Mail, Fax Mail, and Personal Home Page information.

The protection is offered against the following type of attackers: People with access to Web; Other subscribers; MCI personnel; People with access to Subscriber's network; People with access to Subscriber's system; People looking over the shoulder of the Subscriber; and Other systems pretending to be Server(s).

The project implements the security by using the following schemes: Single use TOKENS for login attempts; Validated TOKENS will accompany all transactions;

TOKEN aging to invalidate a TOKEN if it has not been used for ten minutes; TOKEN is associated with the IP Address of the calling machine, so TOKEN stealing is not an easy option; Use of SSL prevents TOKEN or DATA stealing without having physical access to the customer's display; Use of TOKEN in a form analogous to the Netscape Cookie gives us the option to switch to cookies at a later date. Cookies offer us the facility to hide the TOKEN even further into the document for one extra layer of security; and Use of Hostile-IP table to block multiple offenders without detection by them.

In addition to the security implemented by TOKEN as described above, the Web Server(s) are in a Data Management Zone for further low level security. The DMZ security is discussed below.

D. Login Process Figure 42 shows the Login Process. The sequence of events leading to a successful login is: 1. The user requests a connection to www.galileo.mci.com.

2. A server is selected from a set using DNS Round-robin.

3. An HTML Page is sent to the user's browser.

4. The Page checks the browser for JAVA Compliance and displays a welcome message.

5. If the browser is not Java compliant, the process stops with an appropriate message.

6. If the browser is Java compliant, it automatically issues a "GET Login Screen" request to the www.galileo.mci.com server. This request also switches the browser to SSL v2. It will fail if the Browser is not SSL compliant.

7. The Web Server does the following: A. The Web server gets a Single Use Token from its internal Token service.

B. The Web server picks one applet from a large set.

C. The Web server Records the Applet, Token, and Client IP address in a Database.

D. The Web server sends back the Login Screen, with Applet & Token.

8. User fills in the Login Screen fields - User Id and Passcode.

A. The User Id is the user's Directline number (printed on User's Business cards and is in public domain).

B. The Passcode is a Six digit number known only to the User.

9. When the User presses Enter (or clicks on the LOGIN button) the Java Applet sends the UserId, Passcode, Token, and Scrambled Token back. The Scrambling Algorithm is specific to the Applet that was sent in Step 7D.

10. If the browser's IP address is in the Hostile-IP table, the server goes back to Step 7.

11. The Web server authenticates the Login request against what it recorded in Step 7C.

12. If the test is invalid: if this is the third successive failed attempts from the same IP address server records the Address in Hostile-IP table.

13. The server goes back to Step 7.

14. If the test is valid; The server sends a select services screen to the Browser with an embedded Token. The Token is still associated with the Browser's IP address, but it now has an expiration time.

E. Service Selection When the user selects an option from the Service selection screen, the request is accompanied by the Token. The token is validated before the service is accessed, as shown in Figure 43.

F. Service Operation The screens generated by the Application Servers all contain the Token issued to the user when the Login process was started. This Token has an embedded expiration time and a valid source IP Address. All operation requests include this token as a part of the request.

The service requests are sent by the browser as HTML forms, APPLET based forms or plain Hyper Links. In the first two instances, the Token is sent back as a Hidden field using the HTTP-POST method. The Hyper-Links use either the HTTP-GET method with embedded Token or substitute the Cookie in place of a Token. The format of the Token is deliberately chosen to be compatible with this approach.

1. NIDS Server The NIDS server in the system is isolated from the Web Servers by a router-based firewall.

The NIDS server runs the NIDSCOMM and ASCOMM services that allow TCP clients access to databases on the NIDS server. The NIDSCOMM and ASCOMM services do not allow connectivity to databases not physically located on the NIDS Server.

The following databases (C-tree services) on the NIDS server are used by the Welcome Server, Token Server and Profile Management Application Server: 800 PIN 1CALL (this is a partitioned database); 1CALL~TRANS; COUNTRY; COUNTRY~SET; COUNTRY2 (maybe); COUNTRY~CITY (maybe); NPA~CITY; NPACITY~OA300 (maybe); and OP153T00.

In addition to the C Tree services named above the following new C tree services will be defined in the SERVDEF and used only on the NIDS server dedicated to the system: TOKEN; SERVERS;

HOSTILE~IP, TOKEN HOSTS; and SERVER~ENV.

The following descriptions for these databases do not show the filler field required at the first byte of each record, nor do they attempt to show any other filler fields that may be required for structure alignment along the 4-byte boundaries. This omission is made only for clarity.

The numbers in parentheses next to the field definitions are the number of bytes required to hold the field value.

2. TOKEN database service.

The TOKEN database service is accessed by the Token Servers. The primary operations on this service are Create a new record, read a record for a given Token value and update a record for the given Token value.

A separate chron job running on the NIDS Server itself also accesses this database and deletes obsolete records on a periodic basis. This chron job runs every hour. It does a sequential scan of the database and deletes records for expired tokens.

The TOKEN database service contains the TOKEN records. The TOKEN records use a single key (the TOKEN) and have the following fields. 1. Version (1); 2. Use Flag (Single/Multi) (1); 3. Token Value (16); 4. IP Address (16); 5. User Id (16); 6. Time Granted (4); and 7. Time expires (4).

The key field is the Token Value.

3. SERVERS database service.

The Servers Database Service is accessed by the Welcome Server at configuration time. The records in this database contain the following fields: 1. Application Name (16); 2. Application Server Host Name (32); 3. Application Server Domain Name (32); 4. Weight (1); 5. Application Icon File URL (64); and 6 Application Description File URL (64).

The key field is the combination of Application Name, Server Host Name, and Server Domain Name. This database is read by the Welcome Servers sequentially. This database is also accessed by the Web Administrators to Create, Read, Update and Delete records. This

access is via the ASCOMM interface. The Web Administrators use the a HTML form and CGI script for their administration tasks.

4. HOSTILE~IP database service.

This database is accessed by the Welcome servers to create new records or read existing records based on IP address as the key. The read access is very frequent. This database contains the following fields: 1. IP Address (16); 2. Time entered (4); and 3. Time expires (4).

The key field is the IP Address. All three values are set by the Welcome Server when creating this record. If the entry is to be over-ridden, the service doing the over-ride will only be allowed to change the Time expires value to <epoch~start>, thus flagging the entry as over- ride.

This database is also accessed by the Web Administrators to Create, Read, Update, and Delete records. Access is via the ASCOMM interface. The Web Administrators use the HTML form and CGI script for their administration tasks.

Customer Service uses a specially developed tool to access this database and access is allowed only from within the corporate firewall.

A chron job running on the NIDS server also accesses this database and deletes all obsolete records from this database. This job logs all its activity. The log of this job is frequently examined by the Web Administrators all the time.

5. TOKEN~HOSTS database service.

This database service lists IP Addresses of the hosts trusted by the Token Servers. This database is read by the Token Service at configuration time. The records in this database contain the following fields: 1. IP Address (16);

2. Authority (1); 3. Host Name (32); 4. Host Domain Name (32); and 5. Host description (64).

The key field is the IP Address. The Authority binary flag determines the access level. The low access level only allows validate/re-validate commands on an existing TOKEN, the high access level additionally allows Grant and Validate single use TOKEN commands as well.

This database is also accessed by the Web Administrators to Create, Read Update and Delete records. Access is via the ASCOMM interface. The Web Administrators use the HTML form and CGI script for their administration tasks.

6. SERVER~ENV database service.

This database is read by the Welcome and Application servers at startup. It defines the starting environment for these servers. In one embodiment, only one field (and only for the Welcome Servers) is designed to be used. This is expanded in other embodiments.

The records in this database contain the following fields: 1. Sequence Number (4); 2. Application Name (16); 3. Environment Name (32); and 4. Environment Value (64).

The key field is Sequence Number. Environment values may refer to other environment variables by name. The values are evaluated at run time by the appropriate CGI scripts. The Welcome Servers are assigned the pseudo Application Name of WELCOME.

This database is also accessed by the Web Administrators to Create, Read, Update and Delete records. This access is via the ASCOMM interface. The Web Administrators use the HTML form and CGI script for their administration tasks.

7. Chron Job(s) The NIDS Server runs a cleanup chron job. This job is scheduled to run every hour. The main tasks for this job are the following: 1. Scan the HOSTILE~IP database and report on all records. This report contains all records. The aim to track repeat offenders based on this report.

2. Scan the HOSTILE~IP database and report on records with <epoch~time> as their expiration time.

3. Scan the HOSTILE~IP database and delete obsolete records.

4. Scan the TOKEN database and report on all records. This report format will be geared towards traffic reporting rather than scanning each entry.

5. Scan the TOKEN databbase to delete obsolete records.

G. Standards The following coding standards have been developed: 1. HTML Look and Feel standards; 2. Java Look and Feel standards (derived from the HTML look and feel standards, these are the new class libraries used in development to force a common look and feel on the site's pages); and 3. HTML Programming standards.

H. System Administration The system administration tasks require reporting of at least the following System Operating Parameters to the System Administrators: System stats and disk usage with time stamps; Network operating parameters with time stamps; Web page usage and access statistics with time stamps; TOKEN usage statistics; Hostile IP alarms and statistics; The following tools and utilities are on the Servers in DMZ: Time synchronization; Domain Name Servers;

System Log Monitoring; Alarm reporting; and Secure Shell.

The system generates alarms for the following conditions: Incorrect use of TOKENS; Hostile IP table changes; TOKEN Expiration; and Login attempts.

The alarms will be generated at different levels. The Web Servers use the following broad guidelines: 1. The servers run in a root environment.

2. The administrators are able to start a staging server on a non-standard port to test a new (staged) service.

3. The staging server is accessible from Internet during the staging run.

4. The Administrators have the option to move the staging software from staging area to production area with a single command. There are suitable checks to make sure this is not done accidentally.

1. Product/Enhancement A preferred embodiment enables directlineMCI customers additional control over their profile by providing a graphical user interface, and a common messaging system. The capability to access the power of a preferred embodiment exists in the form of a directlineMCI profile and common messaging system. The user is able to modify his account, customizing his application by making feature/functionality updates. The application enables the power of the future capabilities that a preferred

embodiment integration will provide by allowing the user to run his application.

The user is able to access all of his messages by connecting with just one location. FAX, email, page and voice messages will be accessed through a centralized messaging interface.

The user is able to call into the centralized messaging interface through his message center

interface to retrieve messages. A centralized message interface provides the user the capability to manage his communications easily and effectively.

The user interface has two components, the user's application profile and message center.

The interface is accessible through PC software (i.e., PC Client messaging interface), an ARU or a VRU, and a World Wide Web (WWW) Browser. The interface supports the customization of applications and the management of messages.

The feature/functionality requirements for an embodiment will be presented below. The first piece to be described is the ARU interface and its requirements for the user interface.

message management and profile management. Following the ARU requirements, requirements are also provided for the WWW Browser and PC Client interfaces.

J. Interface Feature Requirements (Overview) A front-end acts as an interface between the user and a screen display server in accordance with a preferred embodiment. The user is able to access the system and directly access his profile and messages. The user interface is used to update his profile and to access his messages. The user's profile information and the user's messages may reside in different locations, so the interface is able to connect to both places. Profile and messaging capabilities are separate components of the interface and have different requirements.

Through his interface, the user is able to update his profile in real-time through profile management. The application profile is the front-end to the user account directory, which is where all of the user account information resides in a virtual location. Also, a user is able to manage his messages (voicemail, faxmail, email, pager recall) through his message center.

The message center is the front-end to the centralized messaging database, which is where all of the user's messages may reside, regardless of message content.

Three user interfaces are supported: DTMF access to an ARU or VRU; WWW Browser access to a WWW Site; and PC Client access to a Messaging Server.

From the ARU, the users are able to update their profiles (directlineMCI only), retrieve voicemail messages and pager recall messages, and retrieve message header (sender, subject, date/time) information for faxmail and email messages. Through the PC Client, the user is limited to message retrieval and message manipulation. The WWW Browser provides the user a comprehensive interface for profile management and message retrieval. Through the WWW Browser, the users are able to update their profiles (directlineMCI, Information Services, List Management, Global Message Handling and Personal Home Pages) and retrieve all message types.

1. The User Account Profile The user is able to access account information through the application profile. The application profile provides an intelligent interface between the user and his account information, which resides in the user account directory. The User Account Directory accesses the individual account information of users. Users are able to read and write to the directory, making updates to their accounts. The directory allows search capabilities, enabling customer service representatives to search for a specific account when assisting a customer.

When a customer obtains a phone number, the user account directory reflects the enrollment, and the user is able to access and update features through his user account profile. If a customer withdraws, the user directory will reflect the deactivation, and the service will be removed from the user's application profile.

In summary, the user account directory provides account information for each of the user's services. However, the user account directory is limited to: directlineMCI profile, Information Services profile, Global Message Handling, List Management and Personal Home Page profiles. This information determines the feature/functionality of the user's application and provides the user with the flexibility that is necessary to customize his application, allowing MCI to meet his continuously changing communication needs.

2. The Database of Messages An important feature that is offered is the integration of messages. Messages of similar and dissimilar content are consolidated in one virtual location. Through a call, the

message center provides the user with a review of all of his messages, regardless of content or access.

Through the interface messaging capabilities, the user is also able to maintain an address book and distribution lists.

This message database is a centralized information store, housing messages for users. The message database provides common object storage capabilites, storing data files as objects.

By accessing the message database, users retrieve voicemail, faxmail, email and pager recall messages from a single virtual location. In addition, by using common object storage capabilities, message distribution is extremely efficient.

K. Automated Response Unit (ARU) Capabilities 1. User Interface The ARU interface is able to perform directlineMCI Profile Management, Information Services Profile Management, message retrieval and message distribution. The DTMF access provided through the ARU is applied consistently across different components within the system. For example, entering alphabetic characters through the DTMF keypad is entered in the same manner regardless if the user is accessing Stock Quote information or broadcasting a fax message to a distribution list.

Voicemail Callback Auto Redial provides the capability to prompt for and collect a DTMF callback number from a guest leaving a voicemail and automatically launch a return call to the guest call back number when retrieving messages. Upon completing the callback, the subscriber will be able to return to the same place where they left off in the mailbox.

Music On-Hold provides music while a guest is on-hold.

Park and Page provides a guest an option to page a directlineMCI subscriber, through the directlineMCI gateway, then remain on-hold while the subscriber is paged. The subscriber receives the page and calls their directlineMCI number where they can select to be connected with the guest on hold. Should the subscriber fail to connect a call with the guest, the guest will receive an option to be forwarded to voicemail. If the subscriber does not have voicemail as a defined option, then the guest a final message will be played for the guest.

Note: The guest has the ability to press an option to be forwarded to voicemail at any time while on hold.

Call Screening with Park and Page An embodiment provides the subscriber with functionality for responding to a park and page, the identity of the calling party (i.e., guest).

This provides the subscribers the ability to choose whether they wish to speak to the guest or transfer the guest to voicemail, prior to connecting the call. Specifically, guests are ARU prompted to record their names when they select the park and page option. When the subscriber respond to the park and page, they will hear an ARU prompt stating, "You have a call from RECORDED NAME", then be presented with the option to connect with the calling party or transfer the party to voicemail. If the subscriber does not have voicemail as a defined option, then the guest will be deposited to a final message. The guest also will have the ability to press an option to be forwarded to voicemail at any time while on hold.

Two-way Pager Configuration Control and Response to Park and Page The system also allows a subscriber to respond to a park and page notification by instructing the ARU to route the call to voicemail or final message or continue to hold, through a command submitted by a two-way pager.

Text Pager Support The system allows a subscriber to page a directlineMCI subscriber, through the directlineMCI gateway, and a leave a message to be retrieved by a text pager. Specifically, upon choosing the appropriate option, the guest will be transferred to either the networkMCI Paging or the SkyTel message center where an operator will receive and submitcreate a text- based message to be retrieved by the subscriber's text pager.

Forward to the Next Termination Number The system provides the capability for the party answering the telephone, to which a directlineMCI call has been routed, to have the option to have the call routed to the next termination number in the directlineMCI routing sequence. Specifically, the called party will receive a prompt from the directlineMCI ARU gateway, which indicates that the call has been routed to this number by directlineMCI and providing the called party with the option to receive the incoming call or have the call routed to the next termination number or destination in the routing sequence. The options presented to a called party include: Press an option to accept the call Press an option to send the call to the next termination Let the call time-out (i.e., no action taken) and then proceed to the next termination.

Less Than 2 Second H Reorigination An embodiment also provides the capability to reoriginate an outbound call, from the <BR> <BR> <BR> <BR> directlineMCI gateway, by pressing the pound (&num) key for less than two seconds. Currently, directlineMCI requires the H key to be depressed for two seconds or more before the subscriber can reonginate a call.

L. Message Management 1. Multiple Media Message Notification The subscriber can receive an accounting of current messages across a number of media, to include voicemail, faxmail, email, paging. Specifically, the subscnber will hear an ARU script stating, for example, "You have 3 new voicemail messages, 2 new faxmail messages, and 10 new email messages.

2. Multiple Media Message Manipulation A subscriber is allowed to access the Universal Inbox to perform basic message manipulation, of messages received through multiple media (voicemail, faxmail, email, paging), through the directlineMCI ARU gateway. Subscribers are able to retrieve voicemail messages and pager messages, and retrieve message header (pnority, sender subject, date/time, size) information for faxmail and email messages. In addition, subscribers are able to save, forward or delete messages reviewed from the ARU interface. The forward feature is limited to distributing messages as either voicemails or faxmails. Only voicemail messages can be forwarded as voicemails. Email, faxmail and pager messages can be forwarded as faxmails; however, it may be necessary to convert email and pager messages to a G3 format.

When forwarding messages as faxmails, subscribers have the ability to send messages to distribution lists and Fax Broadcast lists.

3. Text to Speech The system converts text messages, received as email, faxmail or pager messages, into audio, which can be played back through the directlineMCI gateway. Initially, the text-to-speech capability will be limited to message header (pnority, sender, subject, date/time, size) information.

Subscribers are provided the option to select whether they want to hear message headers first and then select which complete message they want to be played. The only message type that does not support a text-to-speech capability for the complete message will be faxmail messages. The capability only exists to play faxmail headers. FAXmail header information includes sender's ANI, date/time faxmail was received and size of faxmail.

4. Email Forwarding to a Fax Machine Subscribers can forward an email, retneved and reviewed through the directlineMCI ARU gateway, to a subscnber-defined termination number. Specifically, the subscriber has the ability to review an email message through the directlineMCI ARU. After reviewing the message, the subscriber receives, among the standard prompts, a prompt requesting whether he would like to forward the email message to a specified termination number or have the

option to enter an impromptu number. Upon selecting this option and indicating the termination number, the email message is converted to a G3 format and transmitted to the specified termination number. Email attachments that are binary files are supported. If an attachment cannot be delivered to the terminating fax machine, a text message must be provided to the recipient that the binary attachment could not be forwarded. Forwarding of emails to a fax machine does not result in the message being deleted from the "universal inbox".

5. Pager Notification of Messages Received A subscriber can receive a pager notification, on a subscriber-defined interval, indicating the number of messages, by message media, that currently reside in the subscriber's "universal inbox". Specifically, the subscriber will have the ability to establish a notification schedule, through the directlineMCI ARU, to receive a pager message which indicates the number of voicemail, faxmail, email and pager messages that reside in the subscriber's "universal inbox".

6. Delivery Confirmation of Voicemail The system provides the subscriber the ability to receive a confirmation voicemail message when a subscriber-initiated voicemail message was not successfully delivered to the terminating party(s).

7. Message Priontization The system provides the guest the ability to assign either regular or urgent priority to a message. When the subscriber receives an accounting of messages, the prioritization will be indicated, and all urgent messages will be indexed before regular messages. This requirement only applies to voicemails, not faxmails. This will require that the "universal inbox" present the proper message priority for directlineMCI voicemails.

M. Information Services Through the ARU interface, users will be able to receive content from information services which are configurable through the WWW Browser interface. Information content will be provided as an inbound service and an outbound service. The information content that is defined through the WWW Browser (i.e., Profile Management) is defined as the inbound information content and will be limited to: Stock Quotes and Financial News Headline News.

Subscribers also have the ability to access additional information content through the ARU interface; however, this information is not configurable through the WWW Browser (i.e., Profile Management). This additional information content will be referred to as outbound information content and will consist of: Stock Quotes and Financial News; Headline News; Weather; Sports News and Scores; Soap Opera Updates; Horoscopes; .Lottery Results; .Entertainment News; and Traveler's Assist.

The configurable parameters of the inbound information content is defined below. Retrieval of outbound information content will support the entry of alphabetic characters through a DTMF keypad. Entering of alphabetic characters must be consistent with the manner that alphabetic characters are entered through DTMF for list management.

Access to Traveler's Assist will be bundled with the other outbound information services such that the subscriber only has to dial a single 800/8XX number. The 800/8XX call may extend to different termination depending upon the information content selected.

N. Message Storage Requirements The message storage requirements are consistent with the message storage requirements defined below.

O. Profile Management directlineMCI Profile Management Subscribers can also review, update and invoke their directlineMCI account profiles. The directlineMCI profile management capabilities through the ARU interface are consistent with the presentation provided through the WWW Browser and support the following requirements: Create new directlineMCI profiles and assign names to the profile; Invoke directlineMCI profiles; Voice annotate directlineMCI profile names; Update existing directlineMCI profiles; Support the rules-based logic of creating and updating directlineMCI profiles (e.g., selection of only one call routing option, like voicemail, will invoke override routing to voicemail; and updates made in one parameter must ripple through all affected parameters, like paging notification); Enable a directlineMCI number; Enable and define override routing number; and Enable and define FollowMe routing.

Enable and define final routing (formerly called alternate routing) to: Voicemail and pager; -Voicemail only; -Pager only; -Final message; Invoke menu routing if two or more of the call routing options (FollowMe, voicemail, faxmail or pager) are enabled; Define the default number for faxmail delivery; Activate paging notification for voicemail; Activate paging notification for faxmail; and Provide guest option to classify voicemails for urgent delivery;

Define call screening parameters for: -Name and ANI; -ANI only; -Name only; and Enable or disable park and page.

P. Call Routing Menu Change The system also provides the capability for subscribers to modify their call routing termination numbers without having to re-enter termination numbers which they do not wish to change. Specifically, the directlineMCI routing modification capability requires the subscriber to re-enter all termination numbers in a routing sequence should they wish to change any of the routing numbers. This capability permits the subscriber to change only the termination numbers they wish to change, and indicate by pressing the "#" key when they do not wish to change a specific number in the routing sequence.

Q. Two-way Pager Configuration Control and Response to Park and Page The system can also enable or disable predefined directlineMCI profiles through a command submitted by a two-way pager.

R. Personalized Greetings The system provides subscribers the ability to review and update the personalized greeting that will be played from the ARU or displayed from their Personal Home Page. Each greeting is maintained separately and customized to the features available through each interface (ARU or Personal Home Page).

S. List Management The system also provides the subscriber the ability to create and update lists, and create a voice annotation name for a list. Fax Broadcast list management capabilities are integrated with directlineMCI list management capabilities to provide a single database of lists. From the ARU interface, subscribers have the ability to review, update, add or delete members on a

list. In addition, subscribers are able to delete or create lists. The ARU interface is able to use the lists to distribute voicemail and faxmail messages.

Access to distribution lists supports alphabetic list names such that lists are not limited to list code names. Entering of alphabetic characters through DTMF to the ARU for list names is consistent with the manner that alphabetic characters are entered through DTMF for Information Services. The List Management requirements are discussed in greater detail below.

In addition to providing message manipulation capabilities, the PC Client also provides an address book and access to lists. The user is able to make modifications to the address book and manage distribution lists for voice, fax, email and paging messages. In one embodiment, lists created or maintained through the PC Client interface are not integrated with lists created or maintained through the WWW Browser or ARU interfaces, but such integration can be implemented in an alternative embodiment. The subscriber is able to send a message to a distribution list from the PC Client. This requires a two-way interface between the PC Client and the List Management database whereby the PC Client can export a comma delimited or DBF formatted file to the database of lists.

The user is able to create and modify recipient address information through his interface PC software. The user is able to record multiple types of addresses in his address book, including 10 digit ANIs, voice mailbox ids, fax mailbox ids, paging numbers and email addresses (MCIMail and Internet). This information should is saved onto the PC. The address information retained on the PC Client is classified and sorted by recipient's name.

T. Global Message Handling From the ARU interface, subscribers are able to define which message types can be accessed from the "universal inbox". The global message handling requirements are consistent with the requirements defined below.

X. INTERNET TELEPHONY AND RELATED SERVICES The discussion thus far has provided an introduction to the Internet, and therefore Internet telephony, but Internet telephony encompasses quite a few areas of development. The following is a summary of Internet telephony, divided into six key areas. The first area consists of access to internet telephony services. This area involves accessing and utilizing the Internet using such mechanisms as satellites, dialup services, TI, T3, DS3, OC3, and OC12 dedicated lines, SMDS networks, ISDN B-channels, ISDN D-channels, multirate ISDN, multiple B-channel bonded ISDN systems, Ethernet, token ring, FDDI GSM, LMDS, PCS, cellular networks, frame relay, and X.25.

The second area involves sharing Internet telephony. Multimedia data can utilize circuit- switched networks quite readily due to the high reliability and throughput potential. Issues include shared data, pushing URL data between parties, data conferencing, shared whiteboarding, resource collaboration, and ISDN user-user signaling.

The third area deals with routing Internet telephony. Issues include the time-of-day, the day- of-week, the day-of-month, and the day-of-year, in addition to geographic points of origin, network point of origin, and time zone of origin. Analysis of routing also includes user data, destination parties, telephone numbers, lines of origin, types of bearer service, presubscribed feature routing, ANI, and IP addresses. Also, VNET plans, range privileges, directory services, and Service Control Points (SCP)s fall into routing Internet telephony.

The fourth category deals with quality of service. Analysis must include switched networks, ISDN, dynamic modifications, internet telephony, RSVP, and redundant network services. In addition, this category includes hybrid Internet/telephony switches, Ethernet features, ISDN features, analog local loops and public phones, and billing for reserved and/or utilized services.

The fifth category is composed of directory services, profiles, and notifications. Examples are distributed directories, finding-me and follow-me services, directory management of telephony, and user interfaces. Calling party authentication security is also included.

Hierarchical and object-oriented profiles exist, along with directory service user profiles, network profile data structures, service profiles, and order entry profiles.

The sixth category consists of hybrid Internet telephony services. Areas include object directed messaging, Internet telephony messaging, Internet conferencing, Internet faxing, information routing (IMMR), voice communications, and intranets (such as those that exist within a company). Other services include operator services, management service, paging services, billing services, wireless integration, message broadcasts, monitoring and reporting services, card services, video-mail services, compression, authorization, authentication.

encryption, telephony application builders, billing, and data collection services.

The seventh category consists of hybrid Internet media services, which include areas of collaborative work which involve a plurality of users. Users can collaborate on Audio, Data and Video. This area includes media conferencing within the Hybrid network. Then there is a broadly related area of Reservations mechanism, Operator-assisted conferencing, and the introduction of content into conferences. The Virtual locations of these conferences will assume importance in the future. The next-generation Chat Rooms will feature virtual conference spaces with simulated Office Environments.

A. System En vironment for internet Media 1. Hardware A preferred embodiment of a system in accordance with the present invention is preferably practiced in the context of a personal computer such as the IBM PS/2, Apple Macintosh computer or UNIX based workstation. A representative hardware environment is depicted in Figure 1A, which illustrates a typical hardware configuration of a workstation 99 in accordance with a preferred embodiment having a central processing unit 10, such as a microprocessor, and a number of other units interconnected via a system bus 12. The workstation shown in Figure 1A includes a Random Access Memory (RAM) 14, Read Only Memory (ROM) 16, an I/O adapter 18 for connecting peripheral devices such as a communication network (e.g., a data processing network) 81, printer 30 and a disk storage unit 20 to the bus 12, a user interface adapter 22 for connecting a keyboard 24, a mouse 26, a speaker 28, a microphone 32, and/or other user interface devices such as a touch screen (not shown) to the bus 12, and a display adapter 36 for connecting the bus 12 to a display device

38. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC System/7 OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

2. Object-Onented Software Tools A preferred embodiment is written using JAVA, C, and the C++ language and utilizes object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP. A need exists for these principles of OOP to be applied to a messaging interface of an electronic messaging system such that a set of OOP classes and objects for the messaging interface can be provided.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An object is a software package that contains both data and a collection of related structures and procedures.

Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules which present an interface that conforms to an object model and which are accessed at run-time through a component integration architecture. A component integration architecture is a set of architectural mechanisms which allow software modules in different process spaces to utilize each other's capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture.

It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

OOP allows the programmer to create an object that is a part of another object. For example the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects

OOP also allows creation of an object that "derived from" another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine "derives from" the object representing the piston engine. The relationship between these objects is called inheritance.

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with them (e.g., number of pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would identify the same functions with the same names, but each type of piston engine may have different/overnding implementations of functions behind the same name. This ability to hide different implementations of a function behind

the same name is called polymorphism and it greatly simplifies communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, our logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are as follows: Objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, countnes in an economics model, or aircraft in an air-traffic-control system.

Objects can represent elements of the computer-user environment such as windows, menus or graphics objects

An object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities.

An object can represent user-defined data types such as time, angles, and complex numbers, or points on the plane.

With this enormous capability of an object to represent just about any logically separable matters, OOP allows the software developer to design and implement a computer program that is a model of some aspects of reality, whether that reality is a physical entity, a process, a system, or a composition of matter. Since the object can represent anything, the software developer can create an object which can be used as a component in a larger software project in the future.

If 90% of a new OOP software program consists of proven, existing components made from preexisting reusable objects, then only the remaining 10% of the new software project has to be written and tested from scratch. Since 90% already came from an inventory of extensively tested reusable objects, the potential domain from which an error could originate is 10% of the program. As a result, OOP enables software developers to build objects out of other, previously built, objects.

This process closely resembles complex machinery being built out of assemblies and sub- assemblies OOP technology, therefore, makes software engineering more like hardware

engineering in that software is built from existing components, which are available to the developer as objects. All this adds up to an improved quality of the software as well as an increased speed of its development.

Programming languages are beginning to fully support the OOP principles, such as encapsulation, inheritance, polymorphism, and composition-relationship. With the advent of the C++ language, many commercial software developers have embraced OOP. C++ is an OOP language that offers a fast, machine-executable code. Furthermore, C++ is suitable for both commercial-application and systems-programming projects. For now, C++ appears to be the most popular choice among many OOP programmers, but there is a host of other OOP languages, such as Smalltalk, common lisp object system (CLOS), and Eiffel. Additionally OOP capabilities are being added to more traditional popular computer programming languages such as Pascal.

The benefits of object classes can be summarized, as follows: Objects and their corresponding classes break down complex programming problems into many smaller, simpler problems.

Encapsulation enforces data abstraction through the organization of data into small, independent objects that can communicate with each other. Encapsulation also protects the data in an object from accidental damage, but allows other objects to interact with that data by calling the object's member functions and structures.

Subclassing and inheritance make it possible to extend and modify objects through deriving new kinds of objects from the standard classes available in the system. Thus, new capabilities are created without having to start from scratch.

Polymorphism and multiple inheritance make it possible for different programmers to mix and match characteristics of many different classes and create specialized objects that can still work with related objects in predictable ways.

Class hierarchies and containment hierarchies provide a flexible mechanism for modeling real-world objects and the relationships among them.

Libraries of reusable classes are useful in many situations, but they also have some limitations. For example: Complexity. In a complex system, the class hierarchies for related classes can become extremely confusing, with many dozens or even hundreds of classes.

Flow of control. A program written with the aid of class libraries is still responsible for the flow of control (i. e., it must control the interactions among all the objects created from a particular library). The programmer has to decide which functions to call at what times for which kinds of objects.

Duplication of effort. Although class libraries allow programmers to use and reuse many small pieces of code, each programmer puts those pieces together in a different way. Two different programmers can use the same set of class libraries to write two programs that do exactly the same thing but whose internal structure (i.e., design) may be quite different, depending on hundreds of small decisions each programmer makes along the way. inevitably, similar pieces of code end up doing similar things in slightly different ways and do not work as well together as they should.

Class libranes are very flexible. As programs grow more complex, more programmers are forced to reinvent basic solutions to basic problems over and over again. A relatively new extension of the class library concept is to have a framework of class libranes. This framework is more complex and consists of significant collections of collaborating classes that capture both the small scale patterns and major mechanisms that implement the common requirements and design in a specific application domain. They were first developed to free application programmers from the chores involved in displaying menus, windows, dialog boxes, and other standard user interface elements for personal computers.

Frameworks also represent a change in the way programmers think about the interaction between the code they write and code written by others. In the early days of procedural programming, the programmer called libraries provided by the operating system to perform certain tasks, but basically the program executed down the page from start to finish, and the programmer was solely responsible for the flow of control. This was appropriate for printing out paychecks, calculating a mathematical table, or solving other problems with a program that executed in just one way.

The development of graphical user interfaces began to turn this procedural programming arrangement inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop which monitors the mouse,

keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use.

Nevertheless, individual pieces of the program written by the developer still call libraries provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control within each piece after it's called by the event loop.

Application code still "sits on top of" the system.

Even event loop programs require programmers to write a lot of code that should not need to be written separately for every application. The concept of an application framework carries the event loop concept further. Instead of dealing with all the nuts and bolts of constructing basic menus, windows, and dialog boxes and then making these things all work together, programmers using application frameworks start with working application code and basic user interface elements in place. Subsequently, they build from there by replacing some of the generic capabilites of the framework with the specific capabilities of the intended application.

Application frameworks reduce the total amount of code that a programmer must write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g., to create or manipulate a data structure).

A programmer writing a framework program not only relinquishes control to the user (as is also true for event loop programs), but also relinquishes the detailed flow of control within the program to the framework. This approach allows the creation of more complex systems that work together in interesting ways, as opposed to isolated programs with custom code being created over and over again for similar problems.

Thus, as explained above, a framework basically is a collection of cooperating classes that make up a reusable design solution for a given problem domain. It typically provides objects that define default behavior (e.g., for menus and windows), and programmers use it by inheriting some of that default behavior and overriding other behavior so that the framework calls application code at the appropriate times.

There are three main differences between frameworks and class libraries: Behavior versus protocol. Class libraries are essentially collections of behaviors that you can call when you want those individual behaviors in your program. A framework, on the other hand, provides not only behavior but also the protocol or set of rules that govern the ways in which behaviors can be combined, including rules for what a programmer is supposed to provide versus what the framework provides.

Call versus override. With a class library, the code the programmer instantiates objects and calls their member functions. It's possible to instantiate and call objects in the same way with a framework (i.e., to treat the framework as a class library), but to take full advantage of a framework's reusable design, a programmer typically writes code that overrides and is called by the framework. The framework manages the flow of control among its objects. Writing a program involves dividing responsibilities among the various pieces of software that are called by the framework rather than specifying how the different pieces should work together.

Implementation versus design. With class libraries, programmers reuse only implementations, whereas with frameworks, they reuse design. A framework embodies the way a family of related programs or pieces of software work. It represents a generic design solution that can be adapted to a variety of specific problems in a given domain. For example, a single framework can embody the way a user interface works, even though two different user interfaces created with the same framework might solve quite different interface problems.

B. Telephony Over The Internet Voice over the Internet has become an inexpensive hobbyist commodity. Several firms are evolving this technology to include interworking with the PSTN. This presents both a challenge and an opportunity for established carriers like MCI and BT especially in the IDDD arena. This discussion explores how a carrier class service could be offered based on this

evolving technology. Of particular interest are ways to permit interworking between the PSTN and the Internet using 1 plus dialing.

The introductory discussion considers the technical requirements to support PC to PC connectivity in a more robust manner than presently offered, in addition to the technical requirements for a PSTN to Internet voice gateway. Consideration is given to how calls can be placed from PCs to a PSTN destination and visa versa. The case of PSTN to PSTN communications, using the Internet as a long distance network is also explored.

It is shown how such services can be offered in a way that will complement existing PSTN services, buffering lower prices for a lower quality of service. At issue in the longer term is the steady improvement in quality for Internet telephony and whether this will ultimately prove competitive with conventional voice services.

1. Introduction In the mid-late 1970s, experiments in the transmission of voice over the Internet were conducted as part of an ongoing program of research sponsored by the US Defense Advanced Research Projects Agency. In the mid-1980s, UNIX-based workstations were used to conduct regular audio/video conferencing sessions, in modest quantities, over the Internet. These experimental applications were extended in the late 1 980s with larger scale, one-way multicasting of voice and video. In 1995 a small company, VocalTec (www.vocaltec.com), introduced an inexpensive software package that was capable of providing two way voice communications between multi-media PCs connected to the Internet. Thus was born a new generation of telephony over the Internet.

The first software package, and its immediate followers, provided a hobbyist tool. A meeting place based on a Internet Relay Chat "room" (IRC) was used to establish point to point connections between end stations for the voice transfer. This resulted in chance meetings, as is common in chat rooms, or a prearranged meeting, if the parties coordinated ahead of time, by email or other means.

a) How it Works A user with a multi-media PC and an Internet connection can add the Internet Telephony capability by loading a small software package. In the case of VocalTec the package makes a connection to the meeting place (IRC server), based on a modified chat server. At the IRC the user sees a list of all other users connected to the IRC.

The user calls another user by clicking on his name. The IRC responds by sending the IP address of the called party. For dial in users of the Internet an IP address is assigned at dial in time and consequently will change between dial in sessions. If the destination is not already engaged in a voice connection, its PC beeps a ring signal. The called user can answer the phone with a mouse click, and the calling party then begins sending traffic directly to the IP address of the called party. A multi-media microphone and speakers built into or attached to the PC are used as a speakerphone. The speaker's voice is digitized, compressed and packetized for transmission across the Internet. At the other end it is decompressed and converted to sound through the PC's speakers.

b) Implications Telephony over the Internet offers users a low cost service, that is distance and border insensitive. For the current cost of Internet access (at low hourly rates, or in some cases unlimited usage for a flat fee) the caller can hold a voice conversation with another PC user connected to the Internet. The called party contributes to the cost of the conversation by paying for his Internet access. In the case that one or both ends are LAN connected to the Internet by leased lines the call is free of additional charges. All of this is in contrast to the cost of a conventional long distance, possibly international, call.

c) Quality of Service The voice quality across the Internet is good, but not as good as typical telephone toll quality.

In addition, there are significant delays experienced during the conversation. Trying to interrupt a speaker in such an environment is problematic. Delay and quality variations are as much a consequence of distance and available capacity as they are a function of compression, buffering and packetizing time.

Delays in the voice transmission are attributable to several factors. One of the biggest contributors to delays is the sound card used. The first sound cards were half duplex and were designed for playback of recorded audio. Long audio data buffers which helped ensure uninterrupted audio playback introduced real time delays. Sound card based delays are being reduced over time as full duplex cards designed for "speakerphone" applications are brought to the market.

Other delays are inherent in the access line speeds (typically 14.4-28.8 kbps for dial-up Internet access) and in the packet forwarding delays in the Internet. Also there is delay inherent in filling a packet with digitized encoded audio. For example, to fill a packet with 90 ms of digitized audio, the application must wait at least 90 ms to receive the audio to digitize. Shorter packets reduce packet-filling delays, but increase overhead by increasing the packet header to packet payload data ratio. The increased overhead also increases the bandwidth demands for the application, so that an application which uses short packets may not be able to operate on a 14.4 kbps dial-up connection. LAN-based PCs suffer less delay, but everyone is subject to variable delays which can be annoying.

Lastly, there are delays inherent in audio codecs. Codec delays can vary from 5 to 30 ms for encoding or decoding. Despite the higher latencies associated with Internet telephony, the price is right, and this form of voice communication appears to be gaining in popularity.

2. IP Phone as a Commercial Service IP telephony technology is here whether the established carriers like it or not. Clearly the use of the Internet to provide international voice calls is a potential threat to the traditional International Direct Distance Dialing (IDDD) revenue stream. Although it may be several years before there is an appreciable revenue impact, it cannot be stopped, except perhaps within national borders on the basis of regulation. The best defense by the carriers is to offer the service themselves in an industrial strength fashion. To do this requires an improved call setup facility and an interface to the PSTN.

Facilitating PC to PC connections is useful for cases in which the voice conversation needs to be conducted during a simultaneous Internet data packet communication, and the parties don't

have access to separate telephone facilities. Dial-up Internet subscribers with only one access circuit might find themselves in that position. Cost considerations may also play a role in dictating the use of PC to PC telephony. The larger use of this technology will occur when the Internet can be used in place of the long distance network to interconnect ordinary telephone hand sets. The number of multi-media Internet connected PCs in the world (estimated at 10 million) is minuscule compared to the number of subscriber lines worldwide (estimated at 680 million). This service is in the planning stages of several companies.

In the sections below we look at each of the end point combinations possible in a full Internet telephony service. The most important aspects relate to the PSTN to Internet gateway capabilities. Of particular interest is the possibility of providing the PSTN caller with one- step dialing to his called party. The one-step dialing solutions discussed below are in the context of the North American numbering plan. There are essentially four cases: 1. PC to PC; 2. PC to PSTN; 3. PSTN to PC; and 4. PSTN to PSTN.

The first case is addressed by today's IP Phone software. The second and third case are similar but not identical and each requires a gateway between the PSTN and the Internet. The last case uses the Internet as a long distance network for two PSTN telephones.

a) PC to PC (1) Directory Service To facilitate PC to PC Internet Telephony a directory service is needed to find the IP address of the called party based on a name presented by the calling party Early internet telephony software utilized a modified internet chat server as a meeting place. More recently, internet telephony software is replacing the chat server with a directory service which will uniquely identify internet telephone users (perhaps by email address). To receive calls, customers would register with the directory service (for a fee, with recurring charges) and would make their location (IP address) known to the directory system whenever they connect to the Internet and want to be available for calls. The best way to accomplish automatic notification is to get agreement between the vendors of IP phone software on a protocol to notify the

directory service whenever the software is started (automatic presence notification). It would also be desirable, as an option, to find a way to automatically invoke the IP phone software when the IP stack is started.

The directory service is envisioned as a distributed system, somewhat like the Internet Domain Name System, for scalability. This is not to imply, necessarily the user(foo.com format for user identification.

Theoretically only the called parties need to be registered. If the calling party is not registered, then the charge for the call (if there is one) could be made to the called party (a collect call). Alternatively, we can insist that the caller also be registered in the directory and billed through that mechanism (this is desirable since we charge for the registration and avoid the complications that collect calls require). A charge for the call setup is billed, but not for the duration, over and above the usual Internet charges. Duration charges already apply to the dial up Internet user and Internet usage charges, both for dial up and dedicated usage, are probably not too far away.

Collect calls from a registered user may be required to meet market demand. A scheme for identifying such calls to the called party must be devised, along with a mechanism for the called party to accept or reject the collect call. The directory service will track the ability of the called software to support this feature by version number (or, alternatively, this could be a matter for online negotiation between the IP telephony software packages).

In the event of collect calls (assuming the caller is not registered), the caller could claim to be anyone she chooses. The directory service will force the caller to take on a temporary "assigned" identity (for the duration of the call) so the called party will know this is an unverified caller. Since IP addresses are not necessarily fixed, one cannot rely on them to identify parties.

(2) Interoperability Nearly all IP phone software packages on the market today use different voice encoding and protocols to exchange the voice information. To facilitate useful connections the directory will store the type and version (and possibly options) of Internet phone software being used.

To make this work effectively software vendors will report this information automatically to

the directory service. This information will be used to determine interoperability when a call is placed. If the parties cannot interoperate, an appropriate message must be sent to the caller. As an alternative, or in addition to registration of software type, a negotiation protocol could be devised to determine interoperability on the fly, but all packages would have to "speak" it.

There is a question of whether translations between IP phone encoding can be performed with acceptable quality to the end user. Such a service could have a duration and or volume fee associated with it which might limit the desirability of its use. Also, after a shake out period we expect only a few different schemes to exist and they will have interoperability, perhaps through an industry agreed lowest common denominator compression and signaling protocol.

So far, all the IP phone software vendors we have contacted are in favor of an Esperanto that will permit interoperability. If this comes to pass the life span of the translation services will be short, probably making them not economically attractive.

We can help the major software vendors seek consensus on a "common" compression scheme and signaling protocol that will provide the needed interoperability. Once the major vendors support this method the others will follow. This is already happening, with the recent announcements from Intel, Microsoft, Netscape, and VocalTec that they will all support the H.323 standard in coming months. This can be automatically detected at call setup time. The directory service would keep track of which versions of which software can interoperate. To facilitate this functionality the automatic notification of presence should include the current software version. This way upgrades can be dynamically noted in the directory service.

Some scheme must also be defined to allow registration information to be passed between software packages so if a user switches packages she is able to move the registration information to the new application. There is no reason to object if the user has two applications each with the same registration information. The directory service will know what the user is currently running as part of the automatic presence notification. This will cause a problem only if the user can run more than one IP phone package at the same time. If the market requires this ability the directory service could be adapted to deal with it. The problem could also be overcome through the use of negotiation methods between interacting IP phone software packages.

(3) Call Progress Signaling If the user is reachable through the directory system, but is currently engaged in a voice connection, then a call waiting message (with caller ID, something which is not available in the PSTN call waiting service) is sent to the called party and a corresponding message is sent back to the caller.

If the user is reachable through the directory system, but is currently not running his voice software (IP address responds, but not the application -- see below for verification that this is the party in question) then an appropriate message is returned to the caller. (As an option an email could be sent to the called party to alert him to the call attempt. An additional option would be to allow the caller to enter a voice message and attach the "voice mail" to the email.

The service could also signal the caller to indicate: busy, unreachable, active but ignored call waiting, etc. Other notification methods to the called party can also be offered, such as FAX or paging. In each case, the notification can include the caller's identity, when known.) Once the directory system is distributed it will be necessary to query the other copies if contact cannot be made based on local information. This system provides the ability to have various forms of notification, and to control the parameters of those forms.

(4) Party Identification A critical question is how will the directory service know that a called party is no longer where she was last reported (i.e., has "gone away"). The dialed in party might drop off the network in a variety of ways (dialed line dropped, PC hung, Terminal Server crashed) without the ability to explicitly inform the directory service of his change in status. Worse yet, the user might have left the network and another user with a voice application might be assigned the same IP address. (This is OK if the new caller is a registered user with automatic presence notification; the directory service could then detect the duplicate IP address. There may still be some timing problems between distributed parts of the directory service.) Therefore, some scheme must exist for the directory service to determine that the customer is still at the last announced location.

One approach to this is to implement a shared secret with the application, created at registration time. Whenever the directory system is contacted by the software (such as

automatic presence notification or call initialization) or attempts to contact the called party at the last known location, it can send a challenge (like CHAP) to the application and verify the response. Such a scheme eliminates the need for announcing "I am no longer here", or wasteful keep alive messages. A customer can disconnect or turn off his IP phone application at any time without concern for notification to the directory system. If multiple IP phone applications are supported, by the directory service, each may do the challenge differently.

(5) Other Services Encrypted internet telephone conversations will require a consensus from the software vendors to minimize the number of encryption setup mechanisms. This will be another interoperability resolution function for the directory service. The directory service can provide support for public key applications and can provide public key certificates issued by suitable certificate authorities.

The user can also specify on the directory service, that his PC be called (dial out) if she is not currently on-line. Charges for the dial out can be billed to the called party, just as would happen for call forwarding in POTS. The call detail record (CDR) for the dial out needs to be associated with the call detail of an entity in the IP Phone system (the called party). Note that this is different than the PC to PSTN case in that no translation of IP encoded voice to PCM is required, indeed the dial out will use TCP/IP over PPP. If the dial out fails an appropriate message is sent back.

The dial out could be domestic or international. It is unlikely that the international case will exist in practice due to the cost. However, there is nothing to preclude that case and it requires no additional functionality to perform.

b) PC to PSTN The PSTN to Internet gateway must support translating PCM to multiple encoding schemes to interact with software from various vendors. Alternatively the common compression scheme could be used once it is implemented. Where possible, the best scheme, from a quality stand point, should be used. In many cases it will the software vendor's proprietary version. To accomplish that, telcos will need to license the technology from selected vendors. Some vendors will do the work needed to make their scheme work on telco platforms.

(1) Domestic PSTN Destination The PC caller needs to be registered to place calls to the PSTN. The only exception to this would be if collect calls from the Internet are to be allowed. This will add complications with respect to billing. To call a PSTN destination the PC caller specifies a domestic E. 164 address. The directory system maps that address to an Internet dial out unit based on the NPA-NXX. The expectation is that the dial out unit will be close to the destination and therefore will be a local call. One problem is how to handle the case where there is no "local" dial out unit. Another problem is what to do if the "local" out dial unit is full or otherwise not available.

Three approaches are possible. One approach is to offer the dial out service only when local calls are possible. A second approach is to send a message back to the caller to inform him that a long distance call must be placed on his behalf and request permission to incur these charges. A third approach is to place the call regardless and with no notification. Each of these cases requires a way to correlate the cost of the dial out call (PSTN CDR) with the billing record of the call originator (via the directory service).

The third approach will probably add to the customer support load and result in unhappy customers. The first approach is simple but restrictive. Most users are expected to be very cost conscious, and so might be satisfied with approach one. Approach two affords flexibility for the times the customer wants to proceed anyway, but it adds complexity to the operation.

A possible compromise is to use approach one, which will reject the call for the reason that no local out dial is available. We could also add an attribute in the call request that means "I don't care if this ends up as a long distance call." In this case the caller who was rejected, but wants to place the call anyway makes a second call attempt with this attribute set. For customers with money to spare, all PSTN calls could be made with that attribute set.

Placing domestic PSTN calls supports the international calling requirement for Internet originated calls from Internet locations outside the US.

(2) International PSTN Destinations Calls to an international PSTN station can be done in one of two ways. First, an international call could be placed from a domestic dial out station. This is not an attractive service since it saves no money over the customer making an international telephone call himself. Second, the Internet can be used to carry the call to the destination country and a "local" dial out can be made there.

This situation is problematic for it must be agreed to by the carrier at the international destination. This case may be viable in one of two ways. Both ways require a partner at the international destination. One option would be to use a local carrier in the destination country as the partner. A second option would be to use an Internet service provider, or some other service provider connected to the Internet in the destination country.

c) PSTN to PC This case appears to be of least interest, although it has some application and is presented here for completeness.

As noted in the PC to PSTN case the PSTN to Internet gateway will need to support translating PCM to multiple encoding schemes to interwork with software from various vendors. The directory service is required to identify the called PC. Automatic notification of presence is important to keep the called party reachable. The PSTN caller need not be registered with the directory service, for caller billing will be based on PSTN information.

The caller has an E.164 address that is "constant" and can be used to return calls as well as to do billing. Presumably we can deliver the calling number to the called party as an indication of who is calling. The calling number will not always be available, for technological or privacy reasons. It must be possible to signal the PC software that this is a PSTN call and provide the E.164 number or indicate that it is unavailable.

The service can be based on charging the calling phone. This can be done as if the Internet were the long distance portion of the call. This is possible with a second dial tone. If an 800 or local dial service is used it is necessary for the caller to enter billing information.

Alternatively a 900 service will allow PSTN caller-based billing. In either case the caller will need to specify the destination 'phone number" after the billing information or after dialing the 900 number.

A major open issue is how the caller will specify the destination at the second dial tone. Only touch tones are available at best. To simplify entry we could assign an E. 164 address to each directory entry. To avoid confusion with real phone numbers (the PSTN to PSTN case) the numbers need to be under directory control. Perhaps 700 numbers could be used, if there are enough available. Alternatively a special area code could be used. Spelling using the touch tone PAD is a less "user friendly" approach.

3. Phone Numbers in the Internet The best approach is to have an area code assigned. Not only will this keep future options open, but it allows for simpler dialing from day one. Given a legitimate area code the PSTN caller can directly dial the E. 164 address of the PC on the Internet. The telephone system will route the call to an MCI POP where it will be further routed to a PSTN-to-Internet voice gateway. The called number will be used to place the call to the PC, assuming it is on-line and reachable. This allows the PSTN caller to dial the Internet as if it were part of the PSTN.

No second dial tone is required and no billing information needs to be entered. The call will be billed to the calling PSTN station, and charges will accrue only if the destination PC answers. Other carriers would be assigned unique area codes and directories should be kept compatible.

For domestically originated calls, all of the billing information needed to bill the caller is available and the intelligent network service functionality for third party or other billing methods is available via the second dial tone.

4. Other Internet Telephony Carriers All this will get more complicated when number portability becomes required. It may be desirable to assign a country code to the Internet. Although this would make domestic dialing more complex (it appears that dialing anything other than 1 plus a ten digit number

significantly reduces the use of the service) it may have some desirable benefits. In any event the assignment of an area code (or several) and the assignment of a country code are not mutually exclusive The use of a country code would make dialing more geographically uniform.

5. International Access It is unlikely that an international call will be made to the US to enter the Internet in the US.

If it happens, however, the system will have enough information to do the caller-based billing for this case without any additional functionality.

Another possibility is that we will (possibly in partnership) set up to handle incoming calls outside the US and enter the Internet in that country to return to the US, or go anywhere else on the Internet. If the partner is a local carrier, then the partner will have the information needed for billing the PSTN caller.

a) Collect Calls PSTN to PC collect calls require several steps. First, the call to the PSTN to Internet gateway must be collect. The collect call could then be signaled in the same way as PC to PC calls. It will be necessary to indicate that the caller is PSTN based and include the calling E.164 address if it is available.

b) PSTN to PSTN The choice of voice compression and protocol scheme for passing voice between PSTN to Internet gateways is entirely under the carrier's control. Various service levels could be offered by varying the compression levels offered. Different charges could associated with each level. The caller would select a quality level; perhaps by dialing different 800 number services first.

(1) Domestic Destination

Neither the calling nor the called parties need be registered with the directory service to place calls across the Internet. The caller dials a PSTN-to-Internet gateway and receives a second dial tone and specifies, using touch tones, the billing information and the destination domestic E.164 address. 900 service could be used as well. The directory service (this could be separate system, but the directory service already has mapping functionality to handle the PC to PSTN dial out case) will be used to map the call to an out dialer to place a local call, if possible. Billing is to the caller and the call detail of the out dial call needs to be associated with the call detail of the inbound caller.

An immediate question is how to deal with the case where the nearest dial out unit to the number called results in a long distance or toll call, as discussed in PC to PSTN case. The situation here is different to the extent that notification must be by voice, and authorization to do a long distance, or toll call dial out must be made by touch tones. In the event of a long distance dial out the Internet could be skipped altogether and the call could go entirely over the PSTN. It is not clear that there is any cost savings by using the Internet in this case.

(2) One Step Dialing The problem is that the destination PSTN number needs to be entered and, somehow, it needs to be indicated that the destination is to be reached via the Internet rather than the conventional long distance network.

This selection criteria can be conveyed according to the following alternatives: 1. Assign a new 10XXX number that is the carrier's Internet.

2. By subscription.

The first method allows the caller to select the Internet as the long distance carrier on a call by call basis. The second method makes the Internet the default long distance network. In the second case a customer can return to the carrier's conventional long distance network by dialing the carrier's 10XXX code.

The first method has the draw back that the caller must dial an extra five digits. Although many will do this to save money, requiring any extra dialing will reduce the total number of users of the service. The second method avoids the need to dial extra digits, but requires a

commitment by the subscriber to predominately use the Internet as his long distance network.

The choice is a lower price with a lower quality of service.

In the PSTN to PSTN case it is possible to consider offering several grades of service at varying prices. These grades will be based on a combination of the encoding scheme and the amount of compression (bandwidth) applied, and will offer lower cost for lower bandwidth utilization.

To signal the grade of service desired three 10XXX codes could be used. By subscription a particular grade would be the default and other service grades would be selected by a 1 0XXX code.

(3) Service Quality The service quality will be measured by two major factors. First, sound quality, the ability to recognize the caller's voice, and second by the delays that are not present in the PSTN.

On the first point we can say that most of the offerings available today provide an acceptable level of caller recognition. Delay, however, is another story. PC to PC users experience delays of a half second to two seconds. As noted in the introduction much of the delay can be attributed to the sound cards and the low speed dial access. In the case of PSTN to PSTN service both these factors are removed.

The use of DSPs in the PSTN to Internet voice gateway will keep compression and protocol processing times very low. The access to the gateway will be at a full 64 kbps on the PSTN side and likely Ethernet on the Internet side. Gateways will typically be located close to the backbone so the router on the Ethernet will likely be connected to the backbone by a T3 line.

This combination should provide a level of service with very low delays. Some buffering will be needed to mask the variable delays in the backbone, but that can likely be kept to under a quarter of a second in the domestic carrier backbone.

The main differentiation of quality of service will be voice recognition which will be related to bandwidth usage. If needed, the proposed IETF Resource reSerVation setup Protocol

(RSVP) can be used to assure lower delay variation, but the need for the added complexity of RSVP is yet to be established. Also, questions remain regarding the scalability of RSVP for large-scale internet telephony.

(4) Costs An open question is whether using the Internet for long distance voice in place of the switched telephone network is actually cheaper. Certainly it is priced that way today, but do current prices reflect real costs? Routers are certainly cheaper than telephone switches, and the 10 kbps (or so) that the IP voice software uses (essentially half duplex) is certainly less than the dedicated 128 kbps of a full duplex